Data Mining Case Studies

Proceedings of the First International Workshop on Data Mining Case Studies

held at the

2005 IEEE International Conference on Data Mining

Edited by

Brendan Kitts, iProspect
Gabor Melli, Simon Fraser University
Karl Rexer, Rexer Analytics

IEEE COMPUTER SOCIETY

Elder Research, Inc.
Data Mining & Pattern Discovery

ISBN 0-9738918-2-3

IEEE

Data Mining Case Studies

Proceedings of the First International Workshop on Data Mining Case Studies

held at the

2005 IEEE International Conference on Data Mining Houston, USA. November 27, 2005

Edited by

Brendan Kitts
iProspect

Gabor Melli
Simon Fraser University

Karl Rexer
Rexer Analytics

Elder Research, Inc.
Data Mining & Pattern Discovery

IEEE
COMPUTER
SOCIETY

IEEE

ISBN 0-9738918-2-3

Contents

Organizers

Chairs

Brendan Kitts, iProspect
Gabor Melli, Simon Fraser University

Prize Committee

Karl Rexer, PhD., Rexer Analytics
John Elder, PhD., Elder Research
Brendan Kitts, iProspect

Program Committee

Gregory Piatetsky-Shapiro, PhD., KDNuggets
Richard Bolton, PhD., KnowledgeBase Marketing, Inc.
Diane Lye, PhD., Amazon
Simeon J. Simoff, PhD., University of Technology Sydney
David Freed, PhD., Exa Corp.
Kevin Hetherington, MITRE Corp.
Parameshvyas Laxminarayan, iProspect
Tom Osborn, PhD., Verism Inc.
Ed Freeman, Washington Mutual
Brendan Kitts, iProspect
Gabor Melli, Simon Fraser University
Karl Rexer, PhD., Rexer Analytics
John Elder, PhD., Elder Research
Martin Vrieze, Harborfreight
Martin Ester, PhD., Simon Fraser University
Kristen Stevensen, iProspect

Sponsors

Elder Research Inc. (ERI)
The Institute of Electrical and Electronics Engineers (IEEE)

Participants

Authors

Emilio Benfenati	Istituto Di Ricerche Farmacologiche
Alan Benson	Hewlett-Packard Company
Robert Ceurvorst	Synovate
Laksminarayan Choudur	Hewlett-Packard Company
Dave DeBarr	Mitre Corporation
Craig DeVault	SAS Corporation
Nelson F. F. Ebecken	Coppe / UFRJ
John Elder	Elder Research Inc.
Timm Euler	University of Dortmund
Alexandre G. Evsukoff	Coppe / UFRJ
Ed Freeman	Washington Mutual
Silvia Figini	University of Pavia
Claudia Galliano	Sky Italy
Paolo Giudici	University of Pavia
Kevin Hetherington	Mitre Corporation
Brendan Kitts	Isobar Communications
Parameshvyas Laxminarayan	iProspect
Benjamin LeBlanc	Isobar Communications
Frank Lemke	KnowledgeMiner Inc.
Kasindra Maharaj	Synovate
Gabor Melli	PredictionWorks / Simon Fraser University
David Montgomery	Poindexter Systems
Johann-Adolf Mueller	Istituto Di Ricerche Farmacologiche
Benjamin J. Perry	iProspect
Carlos André R. Pinheiro	Brasil Telecom
Daniela Polla	Sky Italy
R. Bharat Rao	Siemens Medical
Karl Rexer	Rexer Analytics
Annette Saunders	SAS Corporation
Ariel Sepulveda	Hewlett-Packard Company
Pramod Singh	Hewlett-Packard Company
Charles Thomas	Hewlett-Packard Company
Zach Eyler-Walker	Mitre Corporation

Gold Sponsors

Elder Research Inc.

Elder Research is a leader in the practice of Data Mining -- discovering useful patterns in data and successfully harnessing the information gained. The principals are active researchers in Data Mining, contributing to the literature of this emerging field in books, conferences, and through highly-regarded short courses and training seminars.

IEEE - The Institute of Electrical and Electronics Engineers

The Institute of Electrical and Electronics Engineers is a non-profit, technical professional association of more than 360,000 individual members in approximately 175 countries. The IEEE is a leading authority in technical areas ranging from computer engineering, biomedical technology and telecommunications, to electric power, aerospace and consumer electronics, among others. Through its technical publishing, conferences and consensus-based standards activities, the IEEE produces 30 percent of the world's published literature in electrical engineering, computers and control technology.

Elder Research, Inc.
Data Mining & Pattern Discovery

IEEE COMPUTER SOCIETY

IEEE

Overview

Motivation

From its inception the field of Data Mining has been guided by the need to solve practical problems. This is reflected in the establishment of the Industrial Track at the annual Association for Computing Machinery KDD conference, and practical tutorials at IEEE's Conference on Data Mining. Yet because of client confidentiality restrictions, few articles describe working, real-world success stories. Anecdotally, success stories are one of the most discussed topics at data mining conferences. It is only human to favor the telling of stories. Stories can capture the imagination and inspire researchers to do great things. The benefits of good case studies include:

1. Education: Success stories help to build understanding.
2. Inspiration: Success stories inspire future data mining research.
3. Public Relations: Applications that are socially beneficial, and even those that are just interesting, help to raise awareness of the positive role that data mining can play in science and society.
4. Problem Solving: Success stories demonstrate how whole problems can be solved. Often 90% of the effort is spent solving non-prediction algorithm related problems.
5. Connections to Other Scientific Fields: Completed data mining systems often exploit methods and principles from a wide range of scientific areas. Fostering connections to these fields will benefit data mining academically, and will assist practitioners to learn how to harness these fields to develop successful applications.

The Workshop

It is our pleasure to announce the "Data Mining Case Studies" workshop. This workshop will highlight data mining implementations that have been responsible for a significant and measurable improvement in business operations, or an equally important scientific discovery, or some other benefit to humanity. Data Mining Case Studies organizing committee members reserve the right to contact the deployment site and validate the various facts of the implementation.

Data Mining Case Studies papers have greater latitude in (a) range of topics - authors may touch upon areas such as optimization, operations research, inventory control, and so on, (b) page length - longer submissions are allowed, (c) scope - more complete context, problem and solution descriptions will be encouraged, (d) prior publication - if the paper was published in part elsewhere, it may still be considered if the new article is substantially more detailed, (e) novelty - often successful data mining practitioners utilize well established techniques to achieve successful implementations and allowance for this will be given.

The Data Mining Practice Prize

Introduction

The Data Mining Practice Prize will be awarded to work that has had a significant and quantitative impact in the application in which it was applied, or has significantly benefited humanity. All papers submitted to Data Mining Case Studies will be eligible for the Data Mining Practice Prize, with the exception of members of the Prize Committee. Eligible authors consent to allowing the Practice Prize Committee to contact third parties and their deployment client in order to independently validate their claims.

Award

Winners and runners up can expect an impressive array of honors including

a. Plaque awarded at the IEEE / ICDM conference awards dinner on November 29th, 2005.
b. Prize money comprising $500 for first place, $300 for second place, $200 for third place, donated by Elder Research.
c. Article summaries about each of the deployments to be published in the journal SIGKDD Explorations, which will also announce the results of the competition and the prize winners
d. Awards Dinner with organizers and prize winners.

We wish to thank Elder Research, their generous donation of prize money, incidental costs, time and support, and the IEEE for making our competition and workshop possible.

Improved Cardiac Care via Automated Mining of Medical Patient Records

R. Bharat Rao
Computer-Aided Diagnosis & Therapy Group
Siemens Medical Solutions, USA, Inc
bharat.rao ατ siemens.com

Abstract

Cardiovascular Disease (CVD) is the single largest killer in the world. Although, several CVD treatment guidelines have been developed to improve quality of care and reduce healthcare costs, for a number of reasons, adherence to these guidelines remains poor. Further, due to the extremely poor quality of data in medical patient records, most of today's healthcare IT systems cannot provide significant support to improve the quality of CVD care (particularly in chronic CVD situations which contribute to the majority of costs).

We present REMIND, a Probabilistic framework for Reliable Extraction and Meaningful Inference from Nonstructured Data*. REMIND integrates the structured and unstructured clinical data in patient records to automatically create high-quality structured clinical data. There are two principal factors that enable REMIND to overcome the barriers associated with inference from medical records. First, patient data is highly redundant – exploiting this redundancy allows us to deal with the inherent errors in the data. Second, REMIND performs inference based on external medical domain knowledge to combine data from multiple sources and to enforce consistency between different medical conclusions drawn from the data – via a probabilistic reasoning framework that overcomes the incomplete, inconsistent, and incorrect nature of data in medical patient records.*

This high-quality structuring allows existing patient records to be mined to support guideline compliance and to improve patient care. However, once REMIND is configured for an institution's data repository, many other important clinical applications are also enabled, including: quality assurance; therapy selection for individual patients; automated patient identification for clinical trials; data extraction for research studies; and to relate financial and clinical factors. REMIND provides value across the continuum of healthcare, ranging from small physician practice databases to the most complex hospital IT systems, from acute cardiac care to chronic CVD management, and to experimental research studies. REMIND is currently deployed across multiple disease areas over a total of over 5,000,000 patients across the US.

1. Introduction

Cardiovascular Disease (CVD) is a global epidemic that is the leading cause of death worldwide (17 million deaths) [78]. The World Health Organization estimates that CVD is responsible for 10% of "Disability Adjusted Life Years" (DALYs) lost in low- and middle-income countries and 18% in high-income countries. (The DALYs lost can be thought of as "healthy years of life lost" and indicate the total burden of a disease as opposed to counting resulting deaths.)

Section 2 motivates our research by describing how current technologies are unable to combat the CVD epidemic. We begin by describing the cardiology burden faced today, with an emphasis on the United States, and discuss some of the factors contributing to the further deterioration of the CVD epidemic. A number of CVD treatment guidelines have been developed by health organizations to assist the physician on how to best treat patients with CVD. Yet adherence to these guidelines remains poor, despite studies overwhelmingly showing that adherence to these guidelines reduces morbidity and mortality, improves quality of life, and dramatically reduces healthcare costs.

One of the most promising ways to improve the quality of healthcare is to implement these guidelines within healthcare IT systems. Unfortunately, as we discuss in Section 2, due to the poor quality of healthcare data in medical patient records (the "Data Gap"), most healthcare IT systems are unable to provide significant support for CVD care: this is particularly true in chronic CVD situations which contribute to the majority of costs. Furthermore, this

"Data Gap" is not likely to improve with the introduction of the Electronic Health Record (EHR), and is further hampered by the lack of standards for clinical data, and the fragmented nature of the healthcare IT industry. Medical patient data is typically scattered in multiple sources and most of the information about the clinical context is stored as unstructured free text – these are dictated by physicians at different time points over the continuum of care delivered to the patient. It is important to note that the data is only "poor" from the point of view of automated analysis by computers; it is of high-enough quality for physicians to document and summarize the delivery of healthcare over multiple patient visits with different physicians. Many of the patients we have analyzed already have electronic data documenting their medical histories for more than 5 years (some going back even 20 years). Over time, exponentially increasing electronic data will be available for analysis for more and more patients. Analyzing this data will allow us to improve the healthcare of individual patients and also to mine new population-based knowledge that can be used to develop improved healthcare methodologies.

In Section 3, we introduce our solution for bridging the "Data Gap," the REMIND algorithm for Reliable Extraction and Meaningful Inference from Nonstructured Data. REMIND is a probabilistic framework for automatically extracting and inferring high-quality clinical data from existing patient records – namely, from patient data collected by healthcare institutions in the day-to-day care of patients, without requiring any additional manual data entry or data cleaning. We discuss the business decisions that influenced the design and development of the REMIND platform – namely, the need to rapidly deploy REMIND in diverse healthcare IT situations, for different clinical applications, and for different diseases, and to easily plug in different analysis algorithms for natural language processing and probabilistic inference. In Section 4 we briefly review the details of the REMIND algorithm [59]. Our goal is not to build a solution for a single application (e.g., implement a particular Heart Failure guideline) but to build a general solution that support multiple different applications for different diseases. Although REMIND was initially developed for automated guideline compliance, many other clinical applications are also supported by our solution, both at the individual patient level and the population level. These include automated methods for: therapy selection for individual patients [26]; patient identification for clinical trials; data extraction for research [67]; quality assurance; and relating financial and clinical factors [57].

In Section 5 we describe a number of successful deployments of our solution for the various applications listed above. This section illustrates that the REMIND platform can be deployed on the entire range of healthcare IT systems in use today, from relatively simple physician office systems, to some of the most complex hospital databases in existence. Further, our solution provides value in both chronic and acute care settings; can support all aspects of physician workflow (screening, diagnosis, therapy and monitoring) and healthcare administration; and provide research support, both in academic institutions and for ongoing pharmaceutical and medical device clinical trials [58][62]. The results provided have been rigorously verified by clinicians and scientists. In this paper we have focused solely on cardiac applications from clinical data. REMIND is currently deployed across multiple disease areas on a total of over 5,000,000 patients.

We review related research in the field of medicine and probabilistic inference in Section 6, We discuss some future applications of REMIND in Section 7, and conclude in Section 8 with our thoughts on further research.

2. Motivation

Since 1990, more people have died worldwide from CVD than from any other cause. Clearly CVD is an international crisis; however, since all applications described in this paper are from US healthcare institutions, we focus on the United States.

2.1. CVD in the United States

In the United States, an estimated 70 million people have some form of CVD. CVD accounts for roughly one million deaths per year (38% of all deaths), and is a primary or contributing cause in 60% of all deaths[4][1]. CVD claims as many lives per year as the next 5 leading causes of death *combined*. Unfortunately, a number of trends suggest that the problems of cardiovascular disease will only be exacerbated in the future. First, the aging of the U.S. population will undoubtedly result in an increased incidence of CVD [9]. Second, there is an explosive increase in the number of Americans that are obese or have type 2 diabetes; these conditions result in increased cardiovascular complications.

In addition to being a personal health problem, CVD is also a huge public health problem. In the United States, it is estimated that $394 billion will be spent in 2005 on treatment and management of cardiovascular disease. By comparison, the estimated cost of *all cancers* is $190 billion. By any measure, the burden of CVD is staggering.

Most patients with CVD will never be cured; rather, their disease must be managed. Often, people with CVD will live for 10 or 20 years after initial diagnosis. A significant portion of the costs associated with CVD comes about when the chronic disease is not managed well, and the patient comes to the emergency room of a hospital with an acute disease, such as a heart attack or stroke. This is further exacerbated by the shortage in the number of cardiologists in the United States. Of the approximately 18,000 practicing cardiologists in the US, over 5,000 are above the age of 55, and 400-500 will retire every year, while less than 300 will enter the workforce. This highlights the need to better manage CVD patients after diagnosis – particularly to provide tools to help the overburdened cardiologist improve the quality of care delivered to CVD patients.

2.2. CVD Guidelines

As the problem of CVD has exploded, so has medical knowledge about how to best diagnose and treat it. New diagnostic tests and therapies are constantly being developed. These tests have shown great promise for both improving the quality of life for the CVD patient, and reducing the burden of health care by reducing the incidence of acute episodes. In an attempt to improve the quality of care for patients, national health organizations, such as the American Heart Association (AHA) and the American College of Cardiology (ACC) have created expert panels to review the results of various clinical trials and studies, extract out best practices, and then codify them into a series of *guidelines*. These guidelines attempt to assist the physician on how to best treat patients with CVD. (This process is not unique to cardiovascular disease, but happens in every branch of medicine.)

Recent studies have shown that strict adherence to these guidelines result in improvements at a personal level, including reduced morbidity and mortality and improved quality of life, as well as reduced costs to the overburdened health care system. Based on these studies CMS (the Center for Medicare & Medicaid Services) has begun a series of programs to reward physicians and hospitals who comply with guidelines in an attempt to improve guideline adherence. These "pay-for-performance" schemes are intended to provide a direct financial incentive to healthcare providers – in this case, CMS is working with hospitals to promote the adoption of the heart attack component of the AHA and ACC cardiac treatment guidelines, which recommend that physicians prescribe a medicine called a beta blocker early after an acute heart attack and continue the treatment indefinitely in most patients. Beta blockers are prescription medicines that help protect the heart muscle and make it easier for the heart to beat normally. Despite being well-known, compliance to this guideline in the U.S. is estimated to be below 50%.

There is overwhelming evidence showing the huge benefits of following these guidelines, from the perspective of the patient, physician, hospital, and public health. Yet overall guideline adherence remains woefully low. There are 3 principal factors which contribute to this lack of compliance.

First, in recent years, there has been an explosion in guidelines. In the United States, the National Guideline Clearinghouse (www.guideline.gov) has almost 1000 guidelines for physicians to follow. These guidelines are often modified on a periodic basis, such as every year, in response to new medical knowledge. A quick search on Google or Med-Line for heart failure guidelines returns several hundred references – some heart failure guidelines, with subsequent modifications are defined in [1][2][3][27][28].

Second, with the growing trend of HMOs, and the economic realities of medicine today, physicians are forced to see more and more patients in a limited amount of time. Often, physicians will only average 10-18 minutes per patient, and carry a patient load of 20-30 patients per day.[1]

Third, there are often multiple physicians and nurses who interact with the patient, and there is often poor communication between these health care workers with regards to the patient. In such a hectic and chaotic environment, it is impossible to (manually) consistently and accurately identify and follow the specific guidelines for that patient among

[1] 10-20 minutes per patient appears reasonable, but it includes *all* activities associated with the patient visit, including: reviewing previous patient history; talking with the patient about their symptoms and history; examining the patient; arriving at a diagnosis; ordering additional tests and procedures; determining what drugs the patient is currently taking; prescribing treatment and medication; explaining the diagnosis and treatment to the patient; counseling the patient on the risks and rewards of the therapy; and ordering referrals if needed; this time also include time needed for the physician to record all the details of the patient visit including positive and negative findings, impressions, orders, final instructions, and finally signing off on the patient bill.

the hundreds of ever-changing requirements in use. Unless the proper clinical guideline is identified and followed at the point of care (that is, when the patient is with his physician), it is not useful.

2.3. Electronic Health Records (EHR)

The electronic health record (EHR) is increasingly being deployed within health care organizations to improve the safety and quality of care[20]. Because a guideline is simply a set of eligibility conditions (followed by a set of recommended treatment actions) it appears fairly straightforward to determine guideline eligibility by evaluating a guideline's inclusion and exclusion criteria against an EHR. Unfortunately, as discussed below and later in Section 5.3, even the best EHRs in the world do not fully capture the information needed to support automated guideline evaluation.

Medical patient data in electronic form is of two types: financial data and clinical data. Financial data consists of all the information required to document the physician's diagnoses and the procedures performed, and is collected primarily for the purpose of being reimbursed by the insurance company or the government. Financial data is collected in a highly structured, well-organized, and normalized fashion, because if it were not in this form, the payers would not reimburse the institution or physician. This data can, therefore, be analyzed, dissected, and summarized in a variety of ways using well-established database and data warehousing methods from computer science.

In addition to structured information about patient demographics, this "financial data" also includes standardized patient diagnoses which are classified according to the internationally accepted standards, ICD-9 (International Classification of Diseases, 9th Revision [76]) and ICD-10 [77]. Many of the criteria used to determine if a patient is eligible for (and therefore should be treated according to) a particular guideline, are based upon diagnostic information. Therefore, it appears as if these structured diagnosis codes would be a rich source for data mining, and particularly for determining whether a patient was eligible for a particular treatment guideline.

Unfortunately, these ICD-9 (and ICD-10) codes are unreliable from the *clinical point of view*. Various studies have shown that the clinical accuracy of ICD codes is only 60%-80% [7]; in other words, when an ICD code is assigned, the patient will have that corresponding clinical diagnosis only 60-80% of the time. The principal reason for this is that billing data reflects financial rather than clinical priorities.

In the United States, reimbursement is based primarily on the severity of diagnosis: for example, although the patient treatments for AMI (heart attack) and Unstable Angina (a less severe cardiac illness) are virtually indistinguishable, the former diagnosis code generates twice the reimbursement for the institution. There have been several well-publicized cases, where institutions have received hefty fines for "over-coding" (i.e., assigning higher diagnosis codes than is justified). Alternately, billing codes may be missing, or "under-coded", so that institutions are not accused by insurance companies of fraudulent claims. Furthermore, at least in the US, this coding is done by medical abstractors, who although trained to do this coding, typically lack the medical training to assess the clinical data and arrive at the correct diagnosis.

Clearly, financial data alone is insufficient for any kind of patient-level clinical decision support (including determining guideline eligibility), because the errors will multiply when multiple such diagnoses are jointly needed to make a decision (for instance to determine eligibility for a guideline).

Operational clinical systems have very poor data quality from the standpoint of access and analysis. The structured clinical data in clinical repositories (labs, pharmacy, etc.) is sparse with gaps in data and in time, inconsistent due to variations in terminology, and can be clinically misleading. Key clinical information is stored in unstructured form in the clinical repository, typically as unstructured free text in patient history and physicals, discharge summaries, progress notes, radiology reports, etc. Further, the nature of the relationships within data are not well defined, and causal relationships and temporal dependencies cannot be unearthed without medical knowledge; for example, it may not be immediately clear to which diagnosis a procedure "belongs". Efforts to extract key clinical information based on natural language processing alone have met with limited success [44] – and for even slightly complex decisions like guideline eligibility, reliability is very poor. Simply put, the data in clinical repositories is often messy, and thus only a small fraction of the clinical data is available for analysis.

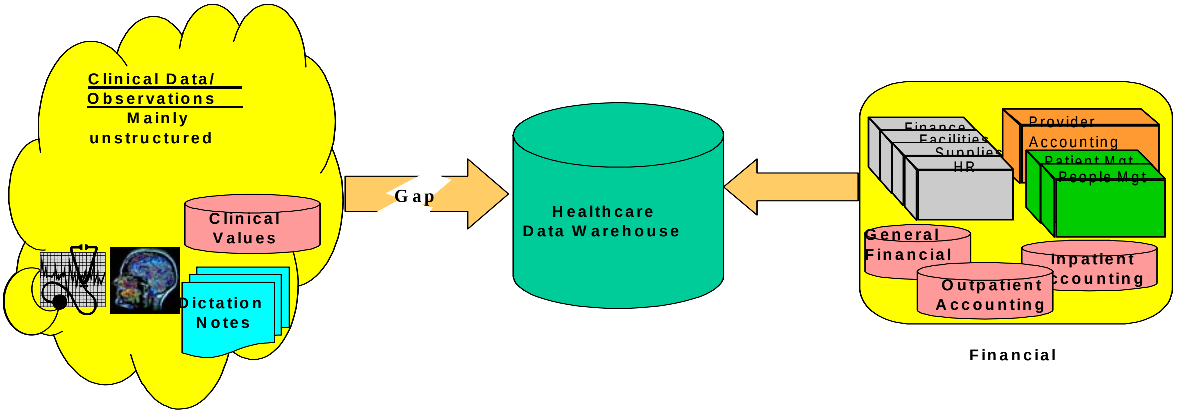


Figure 1: The "Data Gap" in hospital patient records

2.4. The "Data Gap" in medical records

Consider the extremely simple guideline: "*If a patient is admitted with a heart attack, they should be prescribed beta blockers upon discharge*."

In order to assess compliance, it would appear to be sufficient to determine if the patient was admitted with an AMI (acute myocardial infarction or heart attack) and if they were prescribed beta-blockers. Unfortunately, as discussed earlier, even if the patient has an ICD-9 code for an AMI it may not be clinically accurate. The patient may choose to fill a prescription for a beta blocker at a retail pharmacy, so the institution's pharmacy system (if it has one) will have no record of a beta blocker. Most importantly, even if it were possible to determine if the patient did have an AMI this visit and was (or was not) prescribed beta blockers, there are no data fields to determine if beta blockers are contra-indicated, that is, should not be prescribed due to some other reason, such as other medications, complications, or if the patient is known to be allergic to that drug. To receive certification from JCAHO [36], hospitals hire trained nurses to manually extract information from a random sample of 75 emergency room patients about appropriate beta blocker prescription (and a few other very simple guidelines). In short, this cannot be automatically determined using naïve approaches. Figure 1 illustrates the "*Data Gap*" in EHRs that prevents decision-support tools from assisting the physician in providing guideline-directed high-quality care to the patient.

2.5. Automated Patient Data Analysis

Currently there are 3 main ways to perform automated data analysis, discussed below:

1) The most common method, "Limited automated extraction of structured elements only", brings over only the coded financial information (e.g., ICD-9 codes), and loses much of the required clinical information. Further, the coding process has a surprisingly high fraction of errors [57]. Doctors are very pressed for time in the 10-20 minutes they have per patient. If a system alerted a physician about guidelines based on a patient's ICD-9 codes, it would have so many false alerts that the physician would turn it off. (This is not to indicate that billing data is useless. It is used for aggregate level analysis for epidemiological, quality of care, and cost studies,[11][31][48] by hospitals, insurers, the US Dept. of Health Care and CMS. And furthermore, REMIND also leverages this data. The key point is that billing data *alone* is useless for decision support.)

2) "Manual conversion of data by medical experts" leads to high-quality clinical data. But, this is expensive, time consuming, and is only possible for a small subset of patients or at institutions with a strong research focus. It is infeasible for routine clinical use.

3) "Forcing doctors to provide structured input." Currently physicians document their observations as dictated free text, and are extremely efficient at doing so. Taking several minutes (out of the 10-20 m/patient) to additionally fill in specific values in a database can lead to physician resentment, wastes valuable physician time and still leads to missing information (fields may not be provided for all needed information in advance). More clinical data will become available in structured form as EHRs get more accepted. But it will take several years before EHRs will be in routine use for a large fraction of the patient population.

The bottom line is that clinical data is complex, non-uniform and non-homogenous. Automated clinical data analysis of the kind associated with financial data, is almost impossible today. There is a desperate need to create highly-structured clinical data from existing patient records collected by the institution in its day to day practice without requiring any manual data entry or change in physician workflow. Our solution works in the current scenario with poor data quality. However, it is designed to be scalable with respect to the volume and quality of data. REMIND will further benefit as better quality data becomes available, via EHRs or by manual methods.

3. Automated Inference from Medical Patient Records

The "Data Gap" illustrated in Figure 1 explains the inability of automated decision-support systems to assist the physician in providing high-quality care to the patient. As noted earlier, a computer system could not reliably answer from most electronic patient records the question, "Should this patient be given beta-blockers?" However, if the same question is posed to a physician, it is very likely that they (given sufficient time) can answer the question correctly. This means that the information needed to answer the question does exist (or can be inferred) from the electronic patient data, and the "data gap" exists only for computer systems that requires data to be entered in a structured form for analysis.

3.1. Exploiting redundancy in patient records

Patient records typically contain multiple redundant pieces of information about a disease, or indeed any medical situation associated with the patient. For instance, a doctor (or a computer program) could infer that a patient had a particular diagnosis (for example, *is diabetic*) in many different ways and from different data sources:

- Billing codes (ICD-9 of 250.xx for diabetes)
- A transcribed free text dictation that identifies a diagnosis (History and Physical, Discharge summary)
- Symptom (Blood sugar values > 300 in labs)
- Treatment (Insulin or oral anti-diabetic administration in Pharmacy)
- A complication associated with disease (e.g., diabetic nephropathy)
- Other relevant information about the diagnosis (e.g., some steroids elevate blood sugar)

3.2. High-level Requirements

In the above example, the diabetes diagnosis can be inferred from many different data sources and by different methods. For instance, natural language processing could help extract information from free text extraction. A critical component in any successful system must be the use of medical domain knowledge to draw inferences from data.

Fundamentally, any solution must (a) be patient-centric, (b) combine information extracted from all available patient data, and (c) be guided by medical domain knowledge. It follows, therefore, that the system must be able to handle and reason with information in different formats, for instance, doctor's notes in free text and financial, pharmacy, and lab databases from the diabetes example (and in future applications: images, proteomic and genomic data). Further, this information may be contradictory (or indicate the presence or absence of diabetes to varying degrees). Therefore, rather than relying on individual data elements to be extracted correctly, probabilistic reasoning is needed to deal with missing, incorrect, and imprecise information in the clinical repository. Finally, a patient history is not static – symptoms, diagnoses, and treatments may all vary, and temporal inferences will be needed to deal with this added complexity.

There are some additional business drivers to consider. First, as medical guidelines change periodically, the medical knowledge associated with our solution will need to be easily modified. Second, once we have a guideline implemented at one institution, we may wish to implement other guidelines at the same institution with minimal effort. Third, as many institutions implement the same guideline, we may wish to transfer our guideline-solution from one institution to another with minimum effort. Fourth, our solution must scale easily from hundreds of patients to millions of patients. Fifth, our solution must handle the data privacy issues inherent to medical data [72].

Flexibility is a key design requirement – our system must be able to easily incorporate new algorithms to meet the needs of future applications and leverage new technologies, for example, new NLP or probabilistic inference methods. Finally, we should be able to support other decision support applications, both at a patient (e.g., patient identification for clinical trials) and population (e.g., quality assurance) levels.

3.3. System Overview

The REMIND algorithm consists of 3 steps. In the *extraction step*, information is extracted from every part of the patient records in isolation, e.g., from every row in a database table, from every phrase in every sentence. Obviously, several thousand such

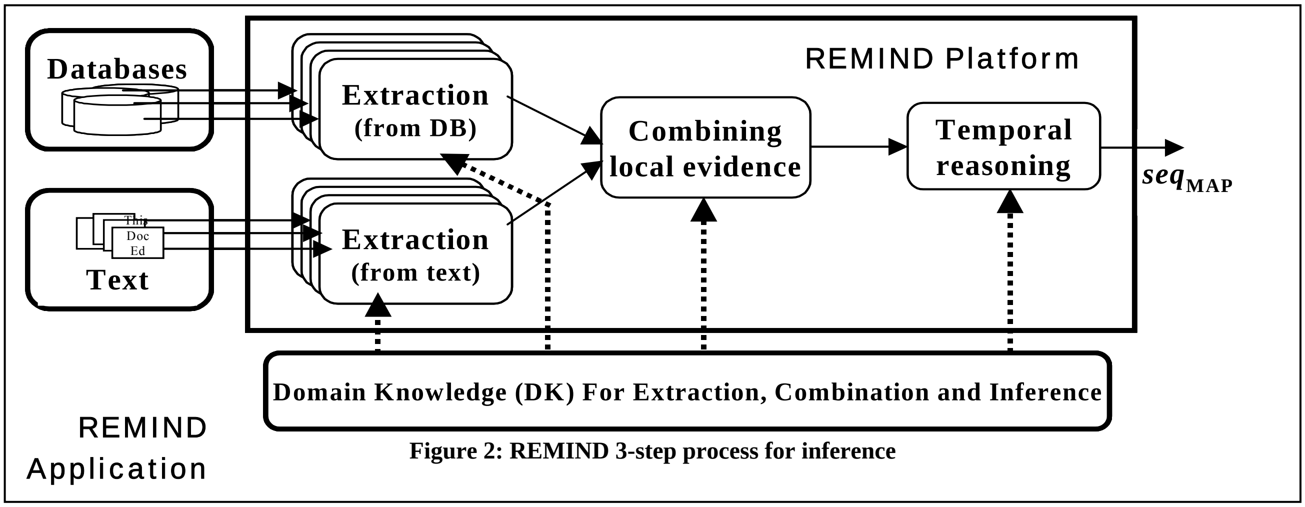


Figure 2: REMIND 3-step process for inference

pieces of information can be extracted from a single patient record, many of which may be incorrect and/or inconsistent, due to errors in the original data or due to the extraction algorithms (for instance, natural language processing is by no means perfect).

In the *combination step*, all observations about a single variable at a single point in time are combined to produce a single observation or a distribution over many different values.

In the *inference step*, different variables are combined across time to infer the values of the variables needed for determining guideline eligibility or compliance. All 3 steps are configured by the medical domain knowledge needed by REMIND to extract the necessary information and arrive at the desired conclusion.

This algorithm is illustrated in Figure 2. The "REMIND Platform" contains all the code needed to implement the 3 steps described above. A key design decision was to implement the medical domain knowledge to drive REMIND as external modules (the "REMIND application") that plugs into the Platform. These REMIND applications are implemented as XML files which configure how the REMIND platform processes data. Therefore, to switch from implementing, say, a heart failure guideline, to recruitment for a coronary artery disease clinical trial, to monitoring radiation treatments for breast cancer, will not require writing any code – each REMIND application will simply consist of a configuration file.

The REMIND domain knowledge to configure the platform is of two types. First, institution-specific domain knowledge describes how the institution's data is organized, where each kind of data is found and under what format, and how to retrieve all data associated with a patient. The second type of knowledge is application-specific. Note that application-specific domain knowledge can be transferred easily from one institution to another – minor retuning would be needed to deal with the differences in the types of data and data quality at each institution, but in general the process of moving an application from one institution to another is low-effort. Similarly, once the REMIND platform is configured for a particular institution (i.e., the institution specific domain knowledge has been created) it also relatively straightforward to implement new application-specific domain knowledge upon the existing configuration.

4. The REMIND Algorithm

In this section we describe one specific application of REMIND (Reliable Extraction and Meaningful Inference from Nonstructured Data). Our goal is to infer disease progression; whether a patient has a particular disease at different points in time, and if so what stage (degree of severity) of the disease. Our medical knowledge about the disease includes knowledge about legal disease sequences – for instance, it may be legal to go from stage 0 to stage 1, but not from 1 to 0, and also information about expected transition times (gathered from the medical literature and survival curves from clinical trials) from one disease stage to another.

Our approach to inference with this multi-source data is to model the data as arising from a generative process, and combine prior knowledge about this process with observations for a specific patient using Bayesian techniques. Patient data is collected in a medical institution at arbitrary points in time (i.e., not at regular intervals but at patient visits only), and these sampling instants vary from patient to patient. Hence, we model the processes of progression of patients' diseases and the collection of this data as continuous time processes that may be sampled at arbitrary instants. We consider a model wherein a patient has a state (for the disease of interest), and observations about the state and related variables are stored in and may be collected from various data repositories.

4.1. Problem Definition

Let **S** be a continuous time random process taking values in Σ that represents the state of the system; note that **S** may be a combination of multiple variables. Let $\mathbf{T}=\{t_1, t_2, \dots, t_n\}$, where $t_i<t_{i+1}$, be the n "times of interest when **S** has to be inferred. Let $\mathbf{S}_i$ refer to the sample of **S** at time $t_i \in \mathbf{T}$. Note that **T** and n can vary for different realizations of the process.

Let **V** be the set of variables that depend upon **S**. Let **O** be set of all (probabilistic) observations for all variables, $\boldsymbol{v} \in \mathbf{V}$. Let $\mathbf{O}_i$ be the set of all observations "assigned" to $t_i \in \mathbf{T}$; i.e., all observations about variables $v \in \mathbf{V}$ that are relevant for this time-step t_i. Similarly, let $\mathbf{O}_i(v)$ be the set of observations for variable v "assigned" to t_i.

Let *seq* be a random variable in Σ^n; i.e., each realization of *seq* is a specific (legal) sequence < $\mathbf{S}_1$, $\mathbf{S}_2$, .. $\mathbf{S}_n$ >.

In the case when we are interested only in the value of a variable at a point in time (e.g., in the AMI example, we simply wish to know if the patient really had an AMI), our goal is to estimate:

$$V_{MAP} = \arg\max_V P[V \mid O]$$

When we wish to track the patient's progress over time, our goal is to estimate the most likely state sequence, seq_{MAP}, the maximum a-posteriori estimate of *seq* given **O**:

$$seq_{MAP} = \arg\max_{seq} P[seq \mid O]$$

4.2. Overview of Approach

We view **S** as a continuous time Markov process from which we observe non-uniform samples. Our implementation of REMIND assumes that **S** is a stationary Markov process, whereas variables, $v \in \mathbf{V}$ that depend on **S** have conditional distributions (on the parent variable) that are non-stationary. However, our framework can be extended to handle even non-stationary Markov processes.

REMIND's 3-step process that estimates the distribution of the variable of interest V_{MAP} (or seq_{MAP}) is summarized below. Our goal is to extract and combine information from all data sources.

(1) **Extraction** step: observations are gathered from the data sources. These observations provide the basic information about the variables $v \in \mathbf{V}$. Operationally; they are converted into a uniform representation, called *probabilistic observations*. These play the same role as likelihood findings in standard Bayesian reasoning. Note that every observation $o \in \mathbf{O}$ is assumed to be potentially incorrect.

(2) **Combination** step: each observation is assigned to one time of interest, $t_i \in \mathbf{T}$. Then each state, $\mathbf{S}_i$ is estimated from all of the observations $\mathbf{O}_i$.

(3) **Inference** step: the inferences are propagated across time and the posterior probabilities for each variable computed.

These steps are in direct correspondence to the different propagation steps of the belief propagation algorithm, well known in the probabilistic inference literature.

4.3. Extraction of probabilistic observations from data

In this step we *produce probabilistic observations, o_i, from data sources*. Each o_i is drawn entirely from a single piece of information in a data source (e.g., from a phrase in a sentence, or a row in a database), and hence is assumed to be inherently undependable (either due to errors in the data or in the extraction process). An observation o_i is of the form <NAME, DATE, DIST> where NAME is an observed variable $v \in \mathbf{V}$, DATE is the date of the observation, and DIST defines a distribution over all possible values that can be taken by NAME given the observation. REMIND currently does extraction from relational databases and free text. Methods from computational linguistics are used to extract information from free text.

These observations generated from the data sources are meant to encode the *a posteriori* distribution of a variable given the section of the data source that they are extracted from, and are subsequently converted into likelihood findings for computation in the Bayesian Network.

4.4. Combination & Inference

The primary focus is estimating what happened to the system (*e.g.,* disease evolution) across the duration of interest. Hence, a natural abstraction of the problem is to look for the best estimate of the sequence of system states across time, and the maximum *a posteriori* (MAP) estimate is the one whose probability is maximal. Hence, given the observations that we have extracted, we would like to estimate the *a posteriori* probability of each legal state sequence and pick the most probable one. This can be done in two steps, the first of which is combination of observations at a fixed point in time and the propagation of these inferences across time.

We use a Markov Model to estimate the evolution of the patient's state. As the observations about patients are spaced non-uniformly across time, the standard discrete-time Markov approximations are not necessarily justifiable. In order to overcome this shortcoming, we model the process of evolution of the patient state as a continuous-time Markov process from which we get to observe non-uniform samples. More specifically, the parameters we need to model are the dwell time in each state and the transition rates from each state to every other. In our current implementation, we consider the state to be a stationary Markov process whereas the other variables that depend on it can have conditional distributions that are non-stationary. Our framework, however, can be modified to handle even the case of non-stationary state processes.

Each piece of information that is extracted in the previous step is in the form of an *a posteriori* probability of a variable given the small context that it is extracted from. We can thus have multiple such assertions from different parts of the same source and from different sources at any given instant in time. All the assertions about a variable at a given point in time are combined into one assertion in a straightforward manner by using Bayes' theorem (under the assumption that the observations are independent given the variable) as follows:

$$\Pr[seq \mid Obs] \propto \Pr[S_0] \cdot \prod_{i=2}^{n} \Pr[S_i \mid S_{i-1}] \cdot \prod_{i=1}^{n} \Pr[Obs_i \mid S_i]$$

$$\propto \prod_{i=2}^{n} \frac{\Pr[S_i \mid S_{i-1}]}{\Pr[S_i]} \cdot \prod_{i=1}^{n} \Pr[S_i \mid Obs_i]$$

We model the relationships between the set of all variables of interest using a Bayesian Network, which is used to infer the posterior distributions of all the variables at a given point in time given all the information at that time. For inference across time, we may now use a standard dynamic programming based approach (e.g. the Viterbi algorithm [56]).

Because we model the state process as being Markov, we have the following equation that connects the *a posteriori* probability of a sequence of samples of the state process given all the observations to the temporally local *a posteriori* probability of the state given all observations at each time instant.

$$\Pr[v \mid O_i^1(v), \ldots, O_i^k(v)] \propto \Pr[v] \cdot \prod_{j=1}^{k} \Pr[O_i^j(v) \mid v] \propto \frac{\prod_{i=1}^{k} \Pr[v \mid O_i^j(v)]}{\Pr[v]^{k-1}}$$

4.5. Domain Knowledge in REMIND

This includes y the state **S** (the variables we wish to infer), **V** (and the data sources for each variable), institution-specific domain knowledge which describes the institution's data structure and access mechanisms, extraction knowledge (e.g, NLP and database queries), dependencies between **S** and **V**, and the dwell times and transitional probabilities.

Despite the seeming complexity, most of the domain knowledge (DK) in REMIND is fairly simple. The clinical application defines S, the variables in V can be elicited fairly easily, and institution-specific knowledge is a one-time implementation effort across many applications at that institution. DK for extraction can be fairly complex, but we have investigated ways to learn this from data. In other medical Bayesian applications [5][38][47], the actual probability values for the dependencies within S and V are typically a huge bottleneck, and require tremendous fine tuning. Because REMIND leverages data redundancy, our systems works well for a wide range of probability values for inference and extraction [62]. Similarly, we roughly estimate dwell times and transition probabilities from survival curves in medical literature. Experiments in [62] also show that REMIND is also insensitive to variations in the temporal parameters.

That said, obviously a big part of the success of any application is the careful tuning that must be done to ensure success. Because of the nature of the application and its potential impact, even if REMIND is inferring information at very high accuracies, there is often value in further improving the end result. All REMIND applications are validated at multiple institutions before release. (The actual DK constitutes the entire REMIND application and is proprietary.)

5. Real-world deployments of REMIND

Here we describe actual deployments of REMIND in various clinical scenarios. Each deployment is characterized by the following variables:

- Name of Institution
- Acute or clinical setting
- IT System, # of physicians and patients supported.

- Population analyzed (may be subset of total)
- Goal (Additional secondary goals are described in parenthesis.)
- Electronic Data Available – at a minimum this will include Billing, demographics, and transcribed free text. Additional specialized databases and free text specialist reports may be available.

5.1. Process Control for Diabetics with AMI

Institution: University of Pittsburgh Medical Center
Setting: Acute Care
IT System: Large hospital IT system supporting several hundred physicians, multiple specialities, multiple locations, 2 Million patients.
Population Analyzed: ER Patients admitted with a diagnosis of Acute Myocardial Infarction (heart attack) in a 3 year period (3000 patients).
Goal: Guideline compliance: proper monitoring of diabetics who had AMI. (Also, clinical and financial outcomes analysis)
Electronic Data: Billing, demographics, pharmacy DBs and transcribed free text (history & physical, progress notes, discharge summaries, ECG and ultrasound reports) from the Medical Archival System (MARS)

Despite the best quality of care provided to cardiology patients, it is inevitable that some people will still face acute episodes and will be rushed to hospital. One of the most common such problems is an acute myocardial infarction (AMI), or heart attack. In fact, for many people, a heart attack is the first symptom that a patient even has cardiac problems. For proper care of patients, it is important that patients who are brought to an emergency room are first properly diagnosed that they have an AMI. Just as critical is to ensure that patients who are diagnosed with an AMI are also assessed for diabetes, and to ensure that their blood sugar is monitored and treated properly, that is, given proper glycemic control (AMI patients with diabetes have much better outcomes if the diabetes is also treated). However, as discussed earlier, these billing codes are inaccurate from the clinical point of view, and are used primarily for reimbursement. Finally, the issue of whether diabetic patients are treated properly for glycemic control cannot be evaluated from structured data alone, but must be inferred from the clinical record.

To address these issues, a study was conducted with the University of Pittsburgh Medical Center[57]. The main purpose of this study was to answer the following three questions:

1. Did patients who were being admitted to the UMPC Intensive Care Unit (ICU) with AMI really have an AMI?
2. Did these patients, who had an AMI, also have diabetes?
3. If the patient had an AMI and diabetes, were they given proper glycemic control?

To address these questions, data was collected from patients who were admitted to the UPMC ICU with a principal diagnosis of AMI (that is, with a principal ICD-9 billing code of 410.xx) in the year 2001. From over 1000 records, 52 were selected randomly.

Table 1. Accuracy for AMI & Diabetes for 52 patients

Diagnosis	ICD-9 CODES			REMIND		
	FP	FN	Acc	FP	FN	Acc
AMI	0	9	83%	1	2	94%
DM	1	4	90%	0	1	98%

Next, clinical definitions of AMI and diabetes were provided from internationally accepted criteria [71] and coded into REMIND. The diagnosis of AMI depends on the unequivocal presence or absence of a combination of three factors upon which the diagnosis rests: symptoms of cardiac pain, abnormalities in the electrocardiogram (ECG), and enzymes released by injured heart muscle. The degree to which those factors meet criteria, individually and in combination, determine the certainty of the AMI diagnosis (“definite”, “probable”, or “possible”). Next, each factor was further defined. For example, for various enzymes released by injured heart muscle, such as troponin, CPK, and CK-MP, various ranges corresponding to abnormal, equivocal, and normal ranges were defined. Similarly, ECG changes and cardiac pain were further defined. For these two cases, the clinical definitions had to be inferred from free text. Diabetes could be inferred either from mention by the physician in their reports, or from either administration of insulin or other oral agents specific to diabetes, or from the presence of lab records showing 2 random blood sugars above 300 mg/dl. Glycemic control was assessed by monitoring blood sugar levels for these patients in the hospital. REMIND was run on 52 patients to answer the 3 questions.

A physician at UPMC, blinded to the results from REMIND, then reviewed the patient record manually for these 52 patients, and then answered the same 3 questions listed above. In making a determination of AMI and diabetes, the physician looked at the entire patient chart (including portions not available to REMIND), and made a clinical diagnosis. The reason was that some parts of the medical record were not in electronic form, and therefore were inaccessible to REMIND. Therefore, the conclusions reached were independent of the domain knowledge and rules provided to REMIND.

Using the physician reads as ground-truth, Table 1 compares the hospital billing codes with Ground Truth, and also REMIND with ground truth for diagnosis of AMI and Diabetes Mellitus (DM). Whereas the diagnostic accuracy of the coded information is only 83% for AMI and 90% for DM, results based on REMIND are much closer to Ground Truth (90% and 95%, respectively). Of the 52 patients coded as AMI, only 43 actually fit the MONICA criteria for AMI (Definite, Probable or Possible). In comparison, REMIND correctly identifies 8 of the 9 patients with No AMI. Of the 52 patients, 19 had diabetes, based on the Ground Truth. REMIND makes only one diagnostic error, compared to 5 in the ICD-9 codes.

Further, REMIND was able to assess whether a patient was given proper glycemic control. It was found that of the 53 patients analyzed, 13 patients had diabetes. In the critical period 24 hours after admission, of these 13, 6 had excellent control of their blood sugars, 1 had moderate control, 1 had poor control, and surprisingly 5 patients were not assessed for blood sugar at all. For their entire stay, 4 had excellent control of their blood sugars, 5 had moderate control, 3 had poor control, and 1 patient was not measured at all. Note that such clinical assessments of process such as glycemic control would be impossible by just looking at billing codes, and would be extremely time consuming (and expensive) for a physician to perform. For instance, the manual chart review averaged 30 minutes per patient, while REMIND ran in mere seconds over all 1000 patients.

Additional Results from the UPMC Analysis:

As mentioned earlier, one of the advantages of our solution is that once it is implemented on the institution's data, it is very easy to extract additional information to support other clinical applications.

One of the most valuable tools available to the hospital administrator is outcomes analysis – namely, analyzing the available data, slicing and dicing it different ways using different database and OLAP tools, to determine the impact of different variables on outcomes. The problem, however, is that the only available data for analysis is the financial data (with diagnosis and procedure codes), and as we show that analysis can lead to incorrect *clinical* conclusions [57].

Table 2 compares the impact of incorrect coding on two key financial outcomes: Length of Stay (LOS) and Charges. (These are good surrogates for quality of care, because in general patients with better care will have shorter hospital stays, and fewer complications, leading to lower charges.) LOS derived from coded information in all 52 patients coded, as having an AMI is about 0.5 days less than the Ground Truth. (This is because 9 patients who actually don't have an AMI, but have been incorrectly coded as having an AMI, are included in computing the Average LOS) Table 2 shows that using the diagnosis extracted by REMIND achieves much greater accuracy, being only 0.1 days off the truth. Similarly, coded information leads to an underestimation of charges incurred in AMI patients by about $5000, whereas REMIND is only off by $1500.

Table 2. Outcomes on AMI patients

Outcomes	ICD-9 Codes	Ground Truth	REMIND
LOS (days)	7.54	7.93	8.05
Charges ($)	$89673	$94688	$96379

There is an additional subtle problem beyond the under-estimation of poor outcomes. Suppose the administrator is considering hiring a diabetic nurse for the ER for the purpose of providing better treatment to diabetics – the next step would be to analyze the available data to determine exactly how much poorer the outcomes were for AMI diabetics versus non-diabetics, and then determine if the potential impact would justify the resources needed to increase staffing.

Table 3. Impact of diabetes on Financial Outcomes

Outcomes	Patient-type	CODERS	Truth	REMIND
LOS (days)	Diabetics	7.13	11.00	11.67
	Non-diabetics	7.70	6.60	6.60
Charges ($)	Diabetics	70,854	105,100	114,887
	Non-diabetics	97,302	90,175	88,976

Table 3 Shows that the errors in coded information regarding AMI and DM compound the underestimation of both LOS and Charges in diabetics with AMI. Thus, coded information would lead to the conclusion that LOS for diabetics was 0.6 days *less* than for non-diabetics, and charges incurred were *lower* by ~$26,000. In actual fact (Ground Truth), diabetics stayed an average of ~4.5 days

longer, and incurred an additional ~$15,000 in *extra* charges. REMIND was much closer to Ground Truth, correctly identifying that diabetics both stayed longer (by ~5 days), and incurred higher charges (by ~$21,000). Table 3 demonstrates the value of REMIND in correctly identifying specific diagnostic categories of patients for outcomes research. They also show the hazards of plotting cost-saving strategies and resource allocations based purely on electronically coded information. For instance, ground truth (and REMIND) reveals exactly the opposite Conclusion about LOS and Charges for diabetics with AMI. This establishes the utility of REMIND, which paralleled Ground Truth, in correctly identifying and analyzing outcomes in a large cohort.

In conclusion, this study showed that REMIND was able to successfully aid in both diagnosis of AMI and diabetes, and in assessing the quality of care for these patients in at least one aspect (glycemic control) in the acute (ICU) environment. The results showed that diagnosis was significantly superior to the use of structured data (i.e. billing codes), and allowed for fast assessment of process quality that could not be assessed using structured data alone.

In many of the deployments that follow, REMIND is used for clinical applications beyond the primary one described. In the interests of brevity, we restrict the description to the primary application.

5.2. Therapy recommendation for patients at risk for Sudden Cardiac Death (SCD)

Institution: South Carolina Heart Center
Setting: Chronic Cardiac Care
IT System: Physician practice IT system supporting 24 cardiologists, single location, 61,027 patients.
Population: All patients.
Goal: Guideline compliance & Therapy Recommendation: identify patients who are at risk for Sudden Cardiac Death, and assess them for defribillator implantation
Electronic Data: Billing and demographics DBs and transcribed free text (history and physical reports, physician progress notes, and lab reports).

Cardiac patients who have had a prior MI are at risk for sudden cardiac death (SCD). Each year, SCD claims the lives of 300,000 Americans. In 1997, a trial was conducted to study the efficacy of using implantable cardioverter defribillators (ICDs) to help prevent sudden cardiac death [49]. The Multicenter Automatic Defibrillator Implantation Trial II (MADIT II), showed that patients who had a prior MI and had low ventricular function, had their 20 month mortality rate drop from 19.8% to 14.2%, a significant 31% reduction in mortality, when an ICD was implanted. The trial was stopped in 2001, with a recommendation to implant ICDs in these patients [50].

Afterwards, there was a need to rapidly identify patients who met these criteria, and evaluate them for ICD implantation. Ordinarily, this could be done in one of two ways. One approach would be to review several thousand patient records manually to assess whether a patient was eligible for an ICD. This approach would be extremely time-intensive and laborious. Another approach could be to evaluate patients as they come in for regular check-ups with their cardiologist. Unfortunately, this would result in needless deaths as patients would only be evaluated if they had a check-up, not to mention the possibility that this new guideline may not be evaluated among the several hundred that the physician must consider. Therefore, it is critical to rapidly assess whether patients were eligible for ICDs, as every month of delay would result in an increased chance of SCD.

Working with the South Carolina Heart Center (SCHC), we implemented REMIND to identify patients who were eligible for an ICD as per the MADIT II study, and who had not yet received an ICD. A total of 61,027 patients were analyzed from the practice for eligibility of an ICD per MADIT II guidelines. REMIND identified 383 patients of the 61,027 as being eligible for an ICD. The total processing time for REMIND for all 61,027 patients was 5 hours on a Pentium M 1.4 GHz laptop.

These 383 patients were mixed with 383 patients randomly drawn from the rest of the population (i.e., MADIT-II ineligible as per REMIND), and 150 of these patients were randomly re-selected. An electrophysiologist manually reviewed the charts for each of these 150 patients to assess MADIT-II eligibility. The reviewer was blinded to the results of REMIND at the time of this determination.

The concurrence between the REMIND system and the manual chart review for eligibility for MADIT-II trial was 94% (141/150). The sensitivity and specificity of REMIND to identify patients were 99% (69/70) and 90% (72/80) respectively. "*Conclusion: REMIND can automatically identify patients who meet definable clinical guideline inclusion/exclusion criteria with a high degree of accuracy. REMIND could be used to improve quality of care and outcomes for patients at risk for cardiovascular disease.*"[26]

5.3. Guideline Adherence Study for Patients with Non-ST Elevation MI

Institution: Veterans Health Administration (VHA) Hospital, Pittsburgh
Setting: Acute Care
IT System: Large hospital IT system supporting several hundred physicians, multiple specialities, multiple locations, 7 Million patients across the US.
Population Analyzed: ER Patients admitted with a diagnosis of unstable angina or non-ST elevated MI over the last 3 years (1400 patients).
Goal: Guideline compliance with ACC guideline.
Electronic Data: Tremendous amounts of structured and unstructured information (see below).

The Veterans Health Administration (VHA) patient database is universally acknowledged as one of the best (if not the best) databases of clinical information in the world. The VHA database is designed to collect a tremendous amount of clinical information in structured form – in addition to the demographics, diagnosis (ICD-9), laboratory, and pharmacy system, many many additional clinical variables are recorded in structured form. Additionally, the VHA database has a vast store of unstructured free text, including history and physicals, admission and discharge reports, progress notes, specialist reports, nursing evaluations, and radiology, ECG, and ultrasound reports. In fact, the VHA database is being strongly recommended by CMS as a model for future EHRs.

It was expectation that with such a tremendous database, the history of quality of care research, and the diligent efforts of the physicians and nurses to keep it current over the last 20 years, there would be little need for automated REMIND analysis. As expected, the support for automated analysis was significantly better than that at any other institution we have encountered. However, somewhat surprisingly we also found that despite the world-class database and research, the available structured data was ineffective for answering questions about the quality of care and compliance.

As discussed previously, one of the big needs in cardiology is to assess whether patients are being treated properly as per established clinical guidelines. The treatment guideline for patients with a certain type of myocardial infarction, in this case patients with non-ST elevation MI was provided by the ACC [10] (http://www.acc.org/clinical/guidelines/unstable/unstable.pdf.). (Another type of myocardial infarction, MI with ST elevation, is treated differently.)

The main responses to the guideline are to provide medication to the patient. For each patient, one must select the correct set of medications for the patient. There are four broad classes of medication for these patients: aspirin; angiotensin converting enzyme (ACE) inhibitors or angiotensin receptor blockers (ARB); beta blockers; and glycoprotein IIb/IIIa receptor anatognists. For each medication, it is important to figure out if the patient should be taking the drug, and also if a patient has a known contra-indication (allergy) to the drug. For example, ACE or ARBs should only be given to patients with diabetes mellitus, congestive heart failure, left ventricular dysfunction or hypertension. In addition, there are a number of reasons a patient even in these conditions should not be given the medication, such as if the patient is pregnant, has pulmonic or aortic stenosis, renal failure, etc. As one can see, the determination of the appropriateness of each class of medication is quite complex.

The VHA has been conducting a retrospective research study on a population of 1400 patients. A trained research nurse manually extracts the information for about 90 variables from these patients. We implemented domain knowledge within REMIND to extract information for about 80 of these variables, and have compared the results of the extraction with the manual extraction on about 1000 patients.

Table 4. Accuracy of REMIND vs. trained medical nurse for guideline compliance

TREATMENT	ACCURACY (%) N=327	
	REMIND	MANUAL
Aspirin	319 (97%)	314 (96%)
Beta Blockers	319 (97%)	316 (97%)
ACE Inhibitors/ARB	300 (92%)	310 (95%)
Glycoprotein IIb/IIIa Receptor Antagonists	300 (92%)	290 (89%)

In this paper, we present the results of analysis for a sub-population of 327 patients admitted with non-ST elevation MI. These patients were studied to see if they were treated properly for each of these four classes of medications as per the ACC guidelines [10]. For each patient, the patient record was searched to see if the patient was treated properly for each of these four medications by both REMIND and manually with the manual abstraction. For each patient, any disagreement between REMIND and the abstraction was adjudicated manually by a medical expert. If REMIND and the research nurse's extraction agreed, both were assumed to be correct.

Note that the research nurse had access to the entire patient record, which includes information that was not available to REMIND.

REMIND v0.5 took 4.5 hours to extract the values of the 4 variables (see Table 11) for 327 patients using a Pentium M 1.6 GHz laptop. (The current version of REMIND is expected to be faster by about 2-3 orders of magnitude.) The medical abstractor took 176 hours to complete the analysis manually for the same variables [67].

Table 4 compares the accuracy of REMIND and manual abstraction for each of the 327 patients. That is, for each patient, this analysis showed what percent of patients were accurately assessed using REMIND and manual abstraction (using the adjudication as a gold standard). Table 4 shows that REMIND works at least as well as manual abstraction in identifying patients who were treated per guidelines for non-ST elevation MI. Note, that the task is different from that shown in Table 1. There, the task was to extract ICD-9 codes, and the comparison was with abstractors who had no medical knowledge. Here, the task is to extract clinical information, and we compare REMIND to a trained nurse with expert medical knowledge; this task is much harder because, as discussed earlier, these medical inferences require subtle inferences to be drawn, particularly for determining contra-indications.

In a controlled study like this, it is possible to spend the time to manually review every patient to assess performance. In reality, however, it is impractical to expect a medical expert to spend time to manually review every patient chart to study if the patient was treated properly or not. In this study, only non-ST elevation MI was considered. If one includes the full spectrum of cardiac diseases, including ST elevation MI, heart failure, arrhythmias, etc., then one can easily see how daunting a task it would be to review every chart for compliance. By using a tool like REMIND, it would be possible to review patients with many different conditions. This would enable physicians to ensure that patients were treated properly, and hence improve their conditions dramatically.

The Veterans Health Administration (VHA) operates 172 medical centers, more than 800 ambulatory care clinics, and provides healthcare for 7 million veterans making it the largest integrated health system in the United States. Further, the VHA mandates that the databases in all institutions are identical. This means that now that REMIND can work successfully at VHA Pittsburgh, it should apply to all the VHA medical centers, with no changes in domain knowledge. This can dramatically increase the impact of our system at the VHA.

5.4. Patient Identification for Clinical Trials

Institution: Nebraska Heart Institute (NHI)
Setting: Chronic Care
Physician practice IT system supporting 32 cardiologists, 4-5 locations, 208,000 patients.
Population: All patients.
Goal: Automatically identify patients who are eligible for clinical trials
Electronic Data: Billing and demographics DBs and transcribed free text (history and physical reports, physician progress notes, and lab reports).

Introducing medical advances into practice is a risky endeavor. Clinical Trials are a critical component to managing this risk. Clinical Trials are required for drugs and medical devices (as an aside, including machine learning-based software to perform Computer-Aided Detection[14][21][61]). Each clinical trial has its own inclusion and exclusion criteria (www.clinicaltrials.gov), which are used to identify eligible patients, and then enroll them, for the trial.

Clinical Trials are very expensive. Pharmaceutical companies spend $20 Billion/yr in the US [43]. Patient recruitment is roughly 10% of a trial's costs ($3.6B in 2002 alone). Difficulty in recruiting patients has been identified as the top cause of delay in Clinical Trials [42]. A key factor in determining the length and cost of a trial is the time it takes to identify and sign up eligible patients for a trial. (For a blockbuster drug, every day's delay in releasing the drug, costs the company, $2Million/day [33].)

REMIND has been used at NHI to successfully identify eligible patients for two actively-recruiting clinical trials. Some "recruitability criteria" (beyond the trial criteria) were added into the analysis; there criteria are used by NHI's trial coordinators to further identify which eligible patients are more likely to agree to participate in the trial (depending on the trial, these criteria could include physical fitness, geographic distance from a hospital, age, co-morbidities, etc.) A key metric for success is the *eligibility rate*, namely, the fraction of eligible patients from all patient records examined. Note that eligibility is determined by examining the entire patient record (including non-electronic data). The traditional method for trial recruitment (other than advertising for patients) is manual examination of patient records by the trial coordinators. Automated identification of eligible patients will help NHI to recruit more patients, with less effort, and less time, resulting in faster and cheaper trials.

The first trial is a medical device trial sponsored by a major device manufacturer. This trial has 18 inclusion/exclusion criteria with a target enrollment of 20 patients at NHI. From NHI's population of 208,000, REMIND identified 363 likely eligible patients. Adding the "recruitability criteria," reduced this list to 31 patients. 29 of these patients were then reviewed by the trial coordinators for eligibility (2 patients could not be easily validated as the appropriate records were at an offsite location.) Of the 29, 18 (62%) were confirmed eligible. Of the 11/29 ineligible patients, 5 were determined to be ineligible from (non-electronic) data available to the coordinators, but not to REMIND.

REMIND has performed even better on an actively recruiting drug trial sponsored by a major pharmaceutical company. This trial has 14 inclusion/ exclusion criteria with a target enrollment of 200 patients at NHI. From NHI's population of 208,000, REMIND identified 2,538 likely eligible patients, of which 312 also met the additional recruitability criteria. Of the 286 patients were validated by NHI, 270 (94%) were confirmed eligible by NHI's trial coordinators. Of the 16/286 ineligible patients, 5 were determined to be ineligible from (non-electronic) data not available to REMIND. (In another process, NHI has contacted 215 of these 270 to participate in the trial; 35 of these patients have accepted.)

To put *eligibility rates* of 62% and 94% in perspective, an *eligibility rate* of 10% would be considered extremely good.

5.5. Quality of Care Analysis for Multiple Institutions

Institution: NHI & SCHC (described earlier)
Population: All 270,000 patients from both institutions
Goal: Automatically extract quality of care information for Heart Failure and Amiodarone

Within the realm of CVD, heart failure imposes the heaviest burden on the healthcare system. In the United States, approximately 5 million people have heart failure, with 550,000 more diagnosed each year. Heart failure results in 12-15 million physician office visits and 6.5 million hospital days each year, and accounts for over 50,000 deaths yearly. Heart failure is a chronic disease with no cure. Patients who are diagnosed with heart failure may live for 10-15 years after the initial diagnosis. Many of the hospital stays associated with heart failure occur because of acute incidents that can be avoided if the patient is properly treated and monitored, and several guidelines have been developed to improve the quality of care [16]. To assist these efforts, several leading medical organizations, including the ACC, AHA, and the AMA, have jointly identified key performance metrics to assist with proper monitoring and treatment of heart failure patients. These metrics are designed to assist the cardiologist monitor the health of the patient, and assess whether changes in treatment are needed. In addition, these metrics list key medications that the patient should be taking. The AMA has created PCPI, the Physician Consortium for Practice Improvement, to be responsible to codify and maintain these metrics.

Unfortunately, as described earlier, simply generating a guideline or metric does not guarantee that physicians will follow them. To assist physicians and practices with compliance to these guidelines, REMIND was used on data from two physician practices consisting of a total of 270,000 patients. First, patients with heart failure were identified using both ICD-9 codes as well as by analyzing the physician notes. Then, each of the metrics in the PCPI guidelines were extracted for these heart failure patients.

For example, the PCPI guidelines state that every heart failure patient should have a number of measurements and assessments taken each year, including left ventricular function, blood pressure, signs and symptoms of cardiac volume overload, activity level, etc. Each of these measurements can be done in a number of different ways. For example, left ventricular function can be assessed using various imaging modalities, such as ultrasound, MRI, nuclear medicine, etc. Activity level can be assessed through observation of the patient through one of many simple exercises. Sometimes, there will be explicit data on these, but other times the assessment of these things must be inferred from the physician's dicated notes. In addition, the PCPI guidelines state that patients should be on medications such as beta blockers, ACE or ARB, and Warfarin (for patients who also have atrial fibrillation) unless there are contra-indications to these medications. REMIND was used to assess each of these guidelines at a patient level, and then aggregated to the entire physician practice (for both practices).

A second analysis was done on patients taking a medication called amiodarone. This is an extremely powerful, but toxic, drug used to treat atrial fibrillation, a cardiac condition. In addition to its toxicity, it often can lead to complications in cardiac patients taking other medications. Because of this, it is very important for patients who are taking amiodarone to be monitored periodically (usually every 6 months) for signs of toxicity. The North American Society of Pacing and Electrophysiology (NASPE) has released a set of guidelines for monitoring patients taking Amiodarone [25]. Our system identifies patients who are taking amiodarone, and then within this subset, those patients who are not being treated as per the NASPE guidelines. The goal here is to help reduce the incidence of side-effects due to the toxic nature of Amiodarone.

The previous applications have presented REMIND results at a single institution. Recall that one of the key requirements in REMIND was to allow the same applications to be run at multiple institutions with little or no retuning. NHI and SCHC have very different healthcare IT systems (different vendors, different kinds of data stored, different databases, different data formats), but being cardiology physician practices, have very similar needs regarding quality of care.

REMIND was run at both institutions' data with virtually no change in domain knowledge. Although validation is ongoing for amiodarone, initial validation results indicate that – at least for cardiology practices – domain knowledge developed at one institution (NHI) retains an equivalent level of performance when transferred to another institution (SCHC). This is critical for rapid deployment. We are in the process of expanding our pool to 1,000,000 cardiology practice patients, and plan to offer a suite of quality of care reports and facilitate benchmarking, both to national standards and across institutions.

6. Related Research

From the earliest days of computing, physicians and scientists have explored the use of artificial intelligence systems in medicine [41]. The original hope was the such systems would become physicians in a box, possibly even surpassing systems in diagnostic tasks [12][61]. Today, the research focus has changed from just diagnosis to support the continuum of healthcare via clinical decision support systems (see, [18][75][63][64]). The fundamental goal of such systems is to reduce costs, improve the quality of care and patient outcomes (see [54] for a summary). Although the impact of such systems on a national scale has been muted, the biggest impact has been made by computerized physician order entry systems that have been shown to reduce medication errors and improve patient outcomes. These systems are based entirely on structured data, and alert the prescribing physician about potentially dangerous drug-drug and drug-disease interactions [24][68].

Another long-standing area of computer research in medicine has been the automated interpretation and analysis of medical images [55]. In the recent past, many such systems have moved out of the realm of research labs into clinical practice, mostly as Computer-Aided Diagnosis systems [21][13] that assist the radiologist in identifying potential cancers in medical images [61][14][8]. We are currently expanding the REMIND platform to include images, and are developing therapy-assistance tools that will help the physician make therapeutic decisions, particularly in the treatment of lung cancer.

From the computer science perspective, our work draws heavily on earlier work on Bayesian networks and graphical models (see [29][34] for an overview). Probabilistic networks have been used in biomedicine and health-care have become increasingly popular for handling the uncertain knowledge involved in establishing diagnoses of disease, in selecting optimal treatment alternatives, and predicting treatment outcomes in various different areas. For example, DxPlain [5] is a decision support system which uses a set of clinical findings (signs, symptoms, laboratory data) to produce a ranked list of diagnoses which might explain (or be associated with) the clinical manifestations. DXplain provides justification for why each of these diseases might be considered, suggests what further clinical information would be useful to collect for each disease, and lists what clinical manifestations, if any, would be unusual or atypical for each of the specific diseases. Quick Medical Reference (QMR [47]) is a large probabilistic graphical model which combines statistical and expert knowledge for approximately 600 significant diseases and 4000 findings. In the probabilistic formulation of the model [65] the diseases and the findings are arranged in a bi-partite graph, and the diagnosis problem is to infer a probability distribution for the diseases given a subset of findings. Promedas [38] is a patient-specific diagnostic decision support system which produces a differential diagnosis on the basis of a set of patient findings. It also suggests the most informative tests that may be performed to make the differential diagnosis more precise. Promedas is based on medical expert knowledge encoded into a probabilistic graphical model (a Bayesian network), which serves as the inference engine of the system. These systems all require clinical data to be entered in a structured database.

Combi et al [19] provides an extensive review of temporal reasoning methods in medicine. We briefly list some methods that are similar to REMIND in some aspects. Ngo et al [52] describe a temporal probabilistic reasoning method via context-sensitive model construction. Bellazi et al [6] describe a system that uses a Dynamic Bayesian Network to analyze the blood glucose level of a patient over a time interval. Kayaalp et al [39] use structured information to predict probabilities of survival for ICU patients. Other related research [32][37][40] deals with representing temporal data and enforcing temporal integrity.

As discussed earlier, a fundamental premise of REMIND, is to exploit the redundancy in the medical record. Our initial implementations achieved very high performance despite using very simple methods from computational linguistics. Although Natural Language Processing (NLP) is not the focus of this work, we are leveraging the rich body of research in this area [45]. Consider the falling examples, all drawn from doctors' dictations, that contain the word Aspirin: "Patient is on Aspirin 2 mg daily; Patient was off Aspirin for a while and then resumed; Dr Smith considered Aspirin 2 mg for him; He stopped taking Aspirin post operative; Use of Aspirin 2mg cannot be excluded; Aspirin on Mondays and Wednesdays; He wants to discuss possible contraindications of his Aspirin dose; Dr Jones ruled out Aspirin for him." Clearly simple look-up for the word "Aspirin" will fail to identify all patients currently taking Asprin. Friedman et al [23] discuss the potential of using NLP techniques in the medical domain, and also provides a comparative overview of the state-of-the-art NLP tools applied to biomedical text. [17][23][30] provide a survey of various approaches to information extraction from biomedical text including named entity tagging and extracting relationship between different entities and between different texts. Clinically relevant observations and features can be extracted with much better accuracy since documents (EMRs) do not have to be treated as bag-of-words ignoring their structure and semantics altogether. For instance, Taira et al [69][35] have done research on automatic structuring of radiology reports. Of direct relevance is the analysis of doctors' dictations by Chapman [15] which identifies the 7 most common uses of negation in doctors' dictations. Augmenting our aliases with a general lexical reference [22] or a medical language dictionary (SNOMED [66]) should improve performance. Furthermore text-mining research to identify relevant documents [46][53] may help eliminate irrelevant documents that are mixed in with doctors' dictations. DISCOTEX [51], like REMIND, extracts information from text, and integrates it via data mining. DISCOTEX focuses on learning rules, whereas REMIND uses domain knowledge for data mining. REMIND is implemented so that text extraction and NLP (and better reasoning) methods can be easily plugged into REMIND.

7. Next Steps

Our immediate next step is to incorporate REMIND into the point of care. Initially, REMIND could be used to alert research coordinators when a potentially eligible patient (for their trial) is being seen by another clinician somewhere else in the clinic. Eligible patients are most likely to enroll if approached at the hospital (since all tests, examinations, and paperwork can be completed on-site, instead of making a separate trip, as would be the case if approached on the phone).

REMIND can also provide point of care support to the physician, for instance, by evaluating the patient against all ongoing open trials and guidelines, and flagging the eligible ones. To this, we are installing REMIND on a multi-million patient database for a large academic medical center.

Other interesting applications include disease surveillance, epidemiological studies, bioterrorism surveillance, and outbreak detection. The RODS [70] (Real-time Outbreak and Disease Surveillance) system mines emergency room data (specifically, 7 fields are provided) and can detect early signs of an outbreak, particularly by detecting spikes in ER admissions. Our approach is complementary, based on a more detailed analysis of individual patient data. We also intend to explore pay-for-performance opportunities with CMS and other payers. Medicine is rich with knowledge bases such as taxonomies (LOINC [60], MeSH [73], and RxNORM), controlled vocabularies (SNOMED CT [66]), and ontologies (UMLS [74]). These systems provide reasoning with crisp logic but unable to handle uncertain knowledge and incomplete/imprecise data. REMIND will incorporate these external sources of knowledge into its inference.

8. Conclusions

We conclude by re-stating some key points:

Medical data is highly complex and difficult to analyze. Financial data is well organized but has limited clinical value. Clinical data is very poor from the point of view of automated analysis (the "Data Gap" in Figure 1). Systems that collect high-quality data will become part of routine clinical care, but are unlikely to have a large patient impact in 5-10 years.

Methods based on analyzing a single kind of data, for example, billing data alone, or just text data (with NLP) are unlikely to have much success. Each source of data has its unique limitations, which might be overcome by information from another data source.

Our solution, REMIND, overcomes these problems by exploiting the redundancy in patient data, and combining information from multiple sources based on external medical knowledge. A probabilistic reasoning system performs the actions necessary to infer high-quality clinical data despite the contradictions, errors, and omissions in the data (and the data extracts from the patient record).

Although our system works with poor data and is not an NLP system, better data and better data extraction methods only improve our performance. REMIND is designed to allow multiple analysis algorithms to be plugged into the platform.

Our goal is to build a general framework to perform inference from medical patient data for a variety of applications and diseases. REMIND provides value in different clinical settings for different diseases. Our system has been designed to support quickly adding data from new institutions, and creating new applications (the domain knowledge files).

The key barrier for IT systems to support automated guideline compliance is the lack of high-quality clinical data collected in day-to-day care. Once REMIND automatically extracts this data, then many other applications are enabled, including: trial recruitment, quality assurance, therapy monitoring, etc.

Here we have only discussed cardiac applications of REMIND. REMIND has been used for other disease areas, including cancer, and efforts are underway to combine images with clinical and financial data to improve analysis. REMIND is current deployed on a rapidly growing population of over 5,000,000 patients.

9. Acknowledgements

This system would not have been successful without the guidance of our clinical collaborators who continue to help shape REMIND: Harsha Rao, MD (Univ of Pittsburgh Medical Center), Colin Germond, MD (Cancer Care Ontario, Canada), Venk Gottipaty, MD, Tim Attebery, and Sherry Schutz (South Carolina Heart Center), Sheryl Dodds, RN, Katie Packard, PharmD, RN, Jacque Taylor, RN, and Kathy Smith, RN (Nebraska Heart Institute), Ali Sonel, MD, Chester Good, MD, Lauren Wall, RN, and Alanna Macioce, RN (VA Hospital, Pittsburgh). Most of all, I am grateful for their support during the early phases of REMIND, and willingness to trust in and help refine a paradigm that is totally different from one traditionally followed in medicine.

Any successful commercial product relies on the dedication of a team. I am indebted to my Siemens colleagues: Sathyakama Sandilya, Ph.D., Ingo Schmuecking, MD, William Landi, Ph.D., Sriram Krishnan, Ph.D., Radu Stefan Niculescu, Ph.D., Alok Gupta, PhD, MBA, Ajit Singh, Ph.D., Arun Goel, Geoff Towell, PhD, Prasad Aloni, Michael Greenberg, Romer Rosales, PhD, Balaji Krishnapuram, PhD, Harald Steck, PhD, Abhinay Pandya, the entire SISL team and Narasimha Murthy. Without their diligence, research and inventiveness, REMIND would have never transitioned from concept to product:

10. References

[1] Advisory Council to Improve Outcomes Nationwide in Heart Failure "Consensus recommendations for the management of chronic heart failure." *Am J Cardiol* 1999;83 (2A):1A-38A.

[2] American College of Cardiology/American Heart Association Task Force on Practice Guidelines "Guidelines for the evaluation and management of heart failure. Report of the ACC/AHA Committee on Evaluation and Management of Heart Failure." *J Am Coll Cardiol* 1995;26:1376-98.

[3] American College of Cardiology/American Heart Association Task Force on Practice Guidelines "ACC/AHA Guidelines for the Evaluation and Management of Chronic Heart Failure in the Adult. ACC/AHA Committee to Revise the 1995 Guidelines for the Evaluation and Management of Heart Failure." *J Am Coll Cardiol*. 2001; 38:2101-13.

[4] American Heart Association. "Heart Disease and Stroke Statistics – 2005 Update." Dallas, TX, *American Heart Association*, 2004.

[5] G. O. Barnett, J. J. Cimino, J. A. Hupp, and E. P. Hoffer, "DXplain: an Evolving Diagnostic Decision-Support System", *JAMA*, 1987, Vol. 258(1), pp. 67-74.

[6] Bellazzi, R., Larizza, C., De Nicolao, G., Riva, A., Stefanelli, M. Mining biomedical time series by combining structural analysis and temporal abstractions. *JAMIA* (symposium supplement), vol. 5 (1998), 160-164.

[7] C. Benesch, D. M. Witter Jr, A. L. Wilder, P. W. Duncan, G. P. Samsa, D. B. Matchar, "Inaccuracy of the International Classification of Diseases (ICD-9-CM) in identifying the diagnosis of ischemic cerebrovascular disease.", *Neurology*, 1997, Vol. 49, pp. 660–664.

[8] L. Bogoni, et al, "CAD for Colonography: A Tool to Address a Growing Need", (to appear in) *British Journal of Radiology*.

[9] R. O. Bonow, L. A. Smaha, S. C. Smith Jr, G. A..Mensah, and C. Lenfant, "World Heart Day 2002: The International Burden of Cardiovascular Disease: Responding to the Emerging Global Epidemic", *Circulation*, 2002, Vol. 106, pp. 1602 – 1605.

[10] Braunwald E, Antman EM, Beasley JW, *et al,* "ACC/AHA 2002 guideline update for the management of patients with unstable angina and non-ST-segment elevation myocardial infarctio.: a report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines", *Committee on the Management of Patients With Unstable Angina* 2002.

[11] J. Broderick, T. Brott, R. Kothari, R. Miller, J. Khoury, A. Pancioli, D. Mills, L. Minneci, R. Shukla, "The

Greater Cincinnati/Northern Kentucky Stroke Study: preliminary first-ever and total incidence rates of stroke among blacks.", *Stroke*. 1998, Vol. 29, pp. 415–421.

[12] B.G.Buchanan, E.H.Shortliffe (eds.) "Rule-Based Expert Systems: the MYCIN experiments of the Stanford Heuristic Programming Project," *Addison-Wesley*, Reading MA, 1984.

[13] S. Buchbinder, I. Leichter, R. Lederman, B. Novak, P. Bamberger, M. Sklair-Levy, G. Yarmish, and S. Fields, "Computer-aided Classification of BI-RADS Category 3 Breast Lesions1", in *Radiology*, 2004, Vol. 230, pp. 820-823.

[14] P. Cathier, et al, "CAD for Polyp Detection: an Invaluable Tool to Meet the Increasing Need for Colon-Cancer Screening", in the *Proceedings of the 18th International Congress and Exhibition, Computer Assisted Radiology and Surgery (*CARS), Chicago, USA, June 23-26, 2004, pp. 978-982.

[15] Chapman, W., Bridewell W., Hanbury P., Cooper, G., Buchanan, B.G., "Evaluation of Negation Phrases in Narrative Clinical Reports", in the *Proceedings of the American Medical Informatics Association* (AMIA) Symposium , 2001, pp. 105-109.

[16] Chavey et al, "Guideline for the Management of Heart Failure Caused by Systolic Dysfunction: Part I. Guideline Development, Etiology and Diagnosis" *American Family Physician*, Vol. 64/No. 5 (September 1, 2001)

[17] A. M. Cohen, and W. R. Hersh, "A Survey of Current Work in Biomedical Text Mining", in *Briefings in Bioinformatic*s, March 2005, Vol. 6(1), pp. 57-71.

[18] Enrico Coiera, "Guide to Health Informatics – 2nd Edition", Arnold Publishers, December 2003.

[19] Combi C., Shahar Y., "Reasoning and Temporal Data Maintenance in Medicine: Issues and Challenges", in *Computers in Biology and Medicine*, Vol. 27(5), 1997, pp. 353-368.

[20] Committee on Data Standards for Patient Safety, Board on Health Services, "Key Capabilities of an Electronic Health Record System: Letter Report," *Institute of Medicine of the National Academies*, 2004.

[21] M. Dundar, G. Fung, L. Bogoni, *et al* "A Methodology for Training and Validating a CAD System and Potential Pitfalls", in the *Proceedings of the 18th International Congress and Exhibition, Computer Assisted Radiology and Surgery* (CARS), Chicago, USA, June 23-26, 2004, pp. 1010-1014.

[22] Fellbaum, C., WordNet: An Electronic Lexical Database. *MIT Press*, May 1998.

[23] C. Friedman, and G. Hripcsak, "Natural Language Processing and Its Future in Medicine: Can Computers Make Sense out of Natural Language Text", in *Academic Medicine*, August 1999, Vol. 74(8), pp. 890-895.

[24] W. Galanter et al., "A Trial of Automated Decision Support Alerts for Contraindicated Medications Using Computerized Physician Order Entry," *J. Am. Med. Inform. Assoc*. 2005;12:269-274.

[25] N.Goldsclager *et al* for the North American Society of Pacing and Electrophysiology, "Practical Guidelines for Clnicians Who Treat Patients with Amiodarone", *Arch Intern Med.* 2000; 160:1741-1748

[26] V. Gottipaty, et al, "Automated Identification Of Madit-II Eligible Patients Using Remind Artificial Intelligence Software", in the 6th Scientific Forum on Quality of Care and Outcomes Research in Cardiovascular Disease and Stroke, *American Heart Association* (AHA), Washington DC, May 14-16, 2005.

[27] Heart Failure Society of America, "HFSA practice guidelines. HFSA guidelines for management of patients with heart failure caused by left ventricular systolic dysfunction--pharmacologic approaches." *J Card Fail* 1999;5:357-82

[28] Heart Society of America, "HFSA Practice Guidelines. HFSA Guidelines for Management of Patients with Heart Failure Caused by Left Ventricular Systolic Dysfunction – Pharmacological Approaches." *Pharmacotherapy*. 2000; 20(5):495-522.

[29] Heckerman, D., "A tutorial on learning with Bayesian networks", *Microsoft Research Technical Report*, MSR-TR-95-06, 1996.

[30] L. Hirschman, J. C. Park, J. Tsujii, L. Wong, and C. H. Wu, "Accomplishments and Challenges in Literature Data Mining for Biology", in *Bioinformatics*, December 2002, Vol. 18(12), pp. 1553-1561

[31] R. G. Holloway, D. M. Witter Jr, K. B. Lawton, J. Lipscomb, G. Samsa, "Inpatient costs of specific cerebrovascular events at five academic medical centers. ", *Neurology*, 1996 Vol. 46, pp. 854–860.

[32] Horn, W., Miksch, S., Egghart, G., Popow, C., Paky, F., "Effective Data Validation of High Frequency Data: Time-Point, Time-Interval, and Trend-Based Methods", *Computers in Biology and Medicine*, 1997.

[33] Jaeger K.D., "Drug Pricing and Consumer Costs", *Pres to US Senate Commerce Committee,* April 23, 2004.

[34] Jensen, F.V., "An introduction to Bayesian Networks", UCL Press, 1996.

[35] Johnson, D.B., Taira, R.K, Zhou, W., Goldin, J.G., Aberle, D.R., Hyperad, "Augmenting and visualizing free text radiology reports", *RadioGraphics*, 1998, Vol. 18, pp. 507-515.

[36] Joint Commission on Accreditation of Healthcare Organizations (JCAHO), Website: http://www.jcaho.org.

[37] Kahn, M., Fagan, L., Tu, S., "Extensions to the Time-Oriented Database Model to Support Temporal Reasoning in Expert Medical Systems", in *Methods of Information in Medicine*, 1991, Vol. 30, pp. 4-14.

[38] B. Kappen, W. Wiegerinck, E. Akay, J. Neijt, and A. van Beek, "Promedas: A Clinical Diagnostic Decision Support System in Bayesian Modeling Applications", in the *Proceedings of the 15th Belgian-Dutch Conference on Artificial Intelligence* (BNAIC'), Nijmegen, The Netherlands, October 23-24, 2003. pp. 455-456.

[39] Kayaalp, M., Cooper, G. F., Clermont G., "Predicting ICU Mortality: A Comparison of Stationary and Nonstationary Temporal Models", in *Proceedings of American Medical Informatics Association* (AMIA) Symposium, 2000, pp. 418-422.

[40] Larizza, C., Moglia, A., Stefanelli, M., "M-HTP: A System for Monitoring Heart Transplant Patients", *Artificial Intelligence in Medicine*, 1992, Vol. 4, pp. 111-126

[41] R. S. Ledley, and L. B. Lusted, “Reasoning Foundations of Medical Diagnosis”, in *Science*, 1959, Vol. 130, pp. 9-21.

[42] Lehman Brothers; McKinsey “Parexel Pharmaceutical International Sourcebook, 2000”, *CenterWatch,* 2000.

[43] Lehman Bros, “Pharma Outsourcing Digest” 3/23/01.

[44] C. L. Leibson, J. M. Naessens, R. D. Brown, J. P. Whisnant, “Accuracy of hospital discharge abstracts for identifying stroke.”, *Stroke*, 1994, Vol. 25, pp. 2348–2355.

[45] Manning, C.D., Schutze, H., “Foundations of Statistical Natural Language Processing”, *MIT Press*, Cambridge, Massachusetts.

[46] McCallum A.K., “BOW: A toolkit for statistical language modeling, text retrieval, classification and clustering”, 1996,: http://www.cs.cmu.edu/~mccallum/bow.

[47] R. A. Miller, F. E. Fasarie, and J. D. Mayors, “Quick Medical Reference (QMR) for Diagnostic Assistance”, *MD Computing*, Sept-Oct, 1986, Vol. 3(5), pp. 34-48.

[48] J. B. Mitchell et al, “What role do neurologists play in determining the costs and outcomes of stroke patients?”, *Stroke*, 1996, Vol. 27, pp. 1937–1943.

[49] Moss AJ, Cannom DS, Daubert JP, et al, for the MADIT II Investigators. “Multicenter Automatic Defibrillator Implantation Trial II (MADIT II): design and clinical protocol.” *Ann Noninvasive Electrocardiology* 1999;4:83-91.

[50] Moss AJ, Zareba W, Hall J, et al, for the Multicenter Automatic Defibrillator Implantation Trial II Investigators. Prophylactic implantation of a defibrillator in patients with myocardial infarction and reduced ejection fraction. *New England Journal of Medicine,* 2002;346:877-883

[51] Nahm U.Y., Mooney R.J., “A Mutual Beneficial Integration of Data Mining and Information Extraction”, in *Proceedings of American Association of Artificial Intelligence* (AAAI), 2000, pp. 627-632.

[52] Ngo L., Haddawy P., Krieger R.A., Helwig J., “Efficient Temporal Probabilistic Reasoning via Context-Sensitive Model Construction”, *Computers in Biology and Medicine*, 1997, Vol. 27(5), pp. 453-476.

[53] Nigam K., McCallum A., Thrun S., Mitchell T. “Learning to Classify Text from Labeled and Unlabeled Documents”, in *Proceedings of the 15th Conference of American Association for Artificial Intelligence* (AAAI), 1998, pp. 792-799 .

[54] J. A. Osheroff, E.A. Pifer, J. M. Teich, MD, et al, "Improving Outcomes with Clinical Decision Support: An Implementer’s Guide", by *Health Information & Management Systems Society*, 2005.

[55] K. Preston, Jr., and M. Onoe (Editors), “Digital Processing of Biomedical Images”, *Plenum Press*, New York, l976.

[56] Rabiner R. L., “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, in the *Proceedings of the IEEE*, Vol. 77(2), pp. 257-286.

[57] R. H. Rao, and R. B. Rao, “Quality Assurance through Comprehensive Extraction from Existing (non-structured) Patient Records”, in the *Annual Conference and Exhibition, Healthcare Information and Management Systems Society* (HIMSS), San Diego, California, Feb 9-13, 2003.

[58] R. B. Rao, S. Sandilya, R. S. Niculescu, C. Germond, and A. Goel. “Mining Time-dependent Patient Outcomes from Hospital Patient Records”, in the *Proceedings of American Medical Informatics Association* (AMIA) Annual Symposium, San Antonio, Texas, November 9-13, 2002.

[59] R. B. Rao, S. Sandilya, R. S. Niculescu, C. Germond, and H. Rao. “Clinical and Financial Outcomes Analysis with Existing Hospital Patient Records”, in the *Proceedings of the Ninth ACM SIGKDD International Conference of Knowledge Discovery and Data Mining* (KDD), Washington DC, August 24-27, 2003, pp. 416-425.

[60] Regenstrief Institute "LOINC: Logical Observation Identifiers Names and Codes", http://www.regenstrief.org/loinc/ LONIC Homepage

[61] J. Roehrig, “The Promise of CAD in Digital Mammography”, in the *European Journal of Radiology*, Elsevier, 1999, Vol. 31, pp. 35-39

[62] S. Sandilya, and R. B. Rao, “Continuous-Time Bayesian Modeling of Clinical Data”, in the *Proceedings of the fourth SIAM International Conference on Data Mining* (SDM), Lake Buena Vista, Florida, April 22-24, 2004.

[63] E. H. Shortliffe, “Computer Programs to Support Clinical Decision Making”, in the *Journal of American Medical Association* (JAMA), 1987, Vol. 258, pp. 61-66.

[64] E. H. Shortliffe (Ed.), L. E. Perreault (Ed.), G. Wiederhold (Asso. Ed.), L. M. Fagan (Asso. Ed.), “Medical Informatics: Computer Applications in Health Care and Biomedicine (Health Informatics) – 2nd Edition”, *Springer*; November, 2000.

[65] M. A. Shwe, B. Middleton, D. E. Heckerman, M. Henrion, E. J. Horvitz, and H.P. Lehmann, “Probabilistic Diagnosis Using a Reformulation of the INTERNIST-1/QMR Knowledge Base: I. The Probabilistic Model and Inference Algorithms”, in the *Methods of Information in Medicine*, October 1991, Vol. 30(4), pp.241-255.

[66] SNOMED International "SNOMED Clinical Terms", College of American Pathologists, http://www.nlm.nih.gov/research/umls/rxnorm_main.htmlhttp://www.snomed.org/

[67] A. F. Sonel, et al, “What is the Most Efficient Data Extraction Method for Quality Improvement and Research in Cardiology?: A Comparison of REMIND Artificial Intelligence Software vs. Manual Chart Abstraction for Determining ACC/AHA Guideline Adherence in Non-ST Elevation Acute Coronary Syndromes”, in the *Annual Scientific Session of American College of Cardiology* (ACC 2005), Orlando, Florida, March 6-9, 2005.

[68] J. Spina et al, "Clinical Relevance of Automated Drug Alerts From the Perspective of Medical Providers", *American Journal of Medical Quality* 2005;20:7-14.

[69] Taira R., Soderland S., Jakobovits R., “Automatic Structuring of Radiology Free Text Reports” *RadioGraphics*, 2001, Vol. 21, pp. 237-245

[70] Tsui F-C, et al, “Technical Description of RODS: A Real-time Public Health Surveillance System.” *Journal Am Med Informatics Assoc* 10/5 (Sept/Oct) 399-408, 2003.

[71] H. Tunstall-Pedoe, “The World Health Organization MONICA Project (Monitoring Trends and Determinants in Cardiovascular Disease): A major international collaboration,” *Journal of Clinical Epidemiology*, 1988, Vol. 41, pp. 105-14.

[72] United States Department of Health and Human Services, "Summary of the HIPAA Privacy Rule", http://www.hhs.gov/ocr/hipaa.

[73] United States Library of Medicine, "MeSH: Medical Subject Headings", http://www.nlm.nih.gov/mesh/

[74] United States Library of Medicine, "UMLS: Unified Medical Language System", http://www.nlm.nih.gov/research/umls/

[75] G. Wiederhold, Edward H. Shortliffe, L.M. Fagan, Leslie E. Perreault, Lawrence M. Fagan (editors) "Medical Informatics : Computer Applications in Health Care and Biomedicine (Health Informatics)" *Springer*; 2nd edition, November, 2000.

[76] World Health Organization “Manual of the international statistical classification or diseases, injuries, and causes of death”, *World Health Organization*, Geneva, 1977.

[77] World Health Organization “Report of the international conferences for the Tenth Revision of International Classification of Diseases”, *World Health Organization*, Geneva, 1992.

[78] World Health Organization, “The Atlas of Global Heart Disease and Stroke”, *World Health Organization & Center for Disease Control*, 2004.

The Mining of SAS Technical Support Data

Annette Sanders and Craig DeVault
SAS Institute, Cary, NC.
Annette.Sanders[ατ]*sas.com*

Abstract

Over the last decade, the amount of data that has been collected and stored has increased dramatically. Statistical methods have been, and continue to be, refined and perfected to yield valuable answers to business problems and goals using standard qualitative and quantitative data analysis methods. However, an overwhelming portion of all data collected is actually unstructured text. The evaluation of textual data is a relatively new focus area of statistics.

The Technical Support Division at SAS has not escaped this deluge of data. Although we collect data for every question or problem that we receive in order to manage the delivery of answers and solutions, we have never used the data as a tool for discovering underlying behavior or patterns of software and support issues -- until now. Taking two years of help desk calls we investigated 3,854 entries using SAS Enterprise Miner to see what we could learn from the text portion of the call report. Basic exploration and preprocessing of textual data is accomplished by parsing and accounting for taxonomy before the typical statistical techniques of data mining can begin such as clustering, SVD, rollup terms, various weighting options, dimension reduction techniques. Mining textual information can deliverer strategic insight to suggest improvements to the business process and may lead to the develop of several predictive models to include regression , decision tree analysis, CHAID, neural nets, etc... This paper outlines how we transformed the alpha data into quantifiable metrics which can deliver solid improvements to our bottom line -- measured in satisfaction, speed of answers, and happy renewing customers.

Closing the Gap: Automated Screening of Tax Returns to Identify Egregious Tax Shelters

Dave DeBarr and Zach Eyler-Walker
The MITRE Corporation
{debarr,zach}[ατ]*mitre.org*

Abstract

According to the most recent strategic plan for the United States Internal Revenue Service (IRS), high-income individuals are a primary contributor to the "tax gap," the difference between the amount of tax that should be collected and the amount of tax that actually is collected [1]. This case study addresses the use of machine learning and statistical analysis for the purpose of helping the IRS target high-income individuals engaging in abusive tax shelters. Kernel-based analysis of known abuse allows targeting individual taxpayers, while associative analysis allows targeting groups of taxpayers who appear to be participating in a tax shelter being promoted by a common financial advisor. Unlike many KDD applications that focus on classification or density estimation, this analysis task requires estimating risk, a weighted combination of both the likelihood of abuse and the potential revenue losses.

1. Introduction

This case study focuses on the use of data analysis techniques to identify egregious tax shelters provided by "pass-through" entities to high-income taxpayers. Trusts, partnerships and subchapter S corporations are referred to as "pass-through" entities because tax liabilities for their income are simply passed to their beneficiaries, partners or shareholders respectively. The allocation of gains and losses from a pass-through entity is recorded for each payee using Schedule K-1 [2,3,4].

Here is a brief characterization of the data available for each type of entity:

- For the purposes of this study, a high-income taxpayer is an individual who reports an annual income of $250 thousand or more. For tax year 2003, 1.9 million high-income returns were filed. The IRS maintains over 1,000 variables to describe each of these returns.
- A trust is a financial entity established to allow a trustee to manage property on behalf of another party, called the beneficiary. The most common type of trust is a grantor trust, in which income of the trust is taxed as income of the grantor. For tax year 2003, 3.5 million trust returns were filed with 4.4 million schedule K-1 records. The IRS maintains over 200 variables to describe each of these returns.
- A partnership is a business in which partners share the gains and losses from operating the business. The most common type of partnership involves leasing real estate property. For tax year 2003, 2.5 million partnership returns were filed with 14.5 million schedule K-1 records. The IRS maintains over 100 variables to describe each of these returns.
- A subchapter S corporation, hereafter referred to simply as an S corporation, is an incorporated business that meets the requirements of subchapter S of the "normal" income taxes chapter of the Internal Revenue Code [5]. For tax year 2003, 3.4 million S corporation returns were filed with 5.9 million schedule K-1 records. The IRS maintains over 100 variables to describe each of these returns.

The IRS had stopped transcribing the schedule K-1 pass-through allocations in 1995, but this practice was resumed for tax year 2000 (calendar year 2001) [6]. The MITRE Corporation was asked to investigate possible analysis methods for exploiting the information about relationships between taxpayers and these pass-through entities. The major lines of investigation included visualization of the relationships and data mining to identify and rank possibly abusive tax avoidance transactions.

2. Visualization

The visualization of the relationships between trusts, partnerships, S corporations and taxpayers included both direct payer to payee relationships and indirect payer to payee relationships; e.g. linking a spouse to a primary filer, a sole proprietorship to an owner or a subsidiary to a parent corporation. Compared to having to repeatedly query a database for linked entities or having to manually switch back and forth between paper returns, this was considered a big improvement. The IRS has subsequently instantiated a prototype visualization system for use by both researchers investigating trends and compliance staff reviewing tax returns. This system now has over 200 user accounts.

Figure 1 illustrates an example of the relationships between a high-income taxpayer and his pass-through entities using a graph with directed edges. All sensitive labels have been removed from the graphs in this paper, but in use the nodes are labeled with name and Taxpayer Identification Number (TIN—either an Employer Identification Number or a Social Security Number), and the edges are labeled with the dollar amounts for gains and losses. A diamond represents a trust, an oval represents a partnership and a rectangle represents an S corporation. The rounded rectangles in the bottom-left are the taxpayer (bold border) and his spouse. The parallelogram in the middle is the taxpayer's sole proprietorship. The octagons indicate the presence of additional payees for three of the partnerships. A user can click on nodes or links to review the transcribed line items for the associated entity or relationship. Colors are used to indicate the presence of various attributes; e.g. red links indicate a net loss and black links indicate a net gain for payer to payee links. The width of a link indicates the amount of money being allocated to a payee; i.e. a thicker link indicates a larger magnitude for money.

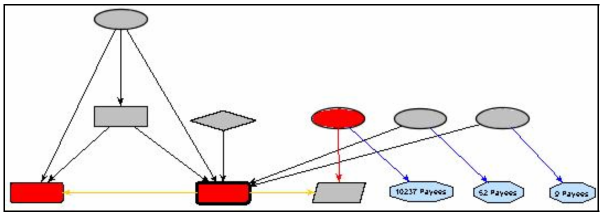


Figure 3: A Taxpayer's Investments

Figure 2 illustrates an example of a prototypical tax shelter with all other relationships for the taxpayer "hidden." This shelter is described in IRS Notice 2000-44 [7] and is commonly referred to as a "Son of BOSS (Bond and Option Sales Strategies)" shelter.

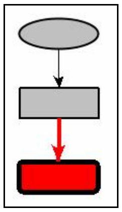


Figure 4: Abusive Son-of-BOSS Tax Shelter

The following description sketches out the salient points of the Son-of-BOSS shelter; however, it is not intended to be a technically accurate description of all related shelter activities. The taxpayer uses a "straddle" to effectively shelter their income:

1. The taxpayer obtains a large $X million gain, often associated with the sale of some asset such as a business.
2. A tax advisor tells the taxpayer he can avoid paying tax on the large gain by exploiting a "loophole" in the tax law. Instead of paying 15-30% tax on the gain, he only needs to pay a smaller fee to the tax advisor.
3. The promoter (tax advisor) sets up a partnership for financial investments, often including himself as a tax matters partner; i.e. the partner who handles tax matters for the partnership (not shown in figure 2).
4. The taxpayer buys call options for $X million; i.e. the option to purchase stock from someone.
5. The taxpayer transfers these call options to the partnership.
6. The taxpayer then sells call options for $X million to someone else; i.e. the option to purchase stock from the taxpayer.
7. The taxpayer ignores the liability of the underwritten call options because the tax advisor claims this is allowable.
8. Upon disposition of the call options, the taxpayer claims an $X million loss to offset his income from the large gain.

The S corporation in figure 2 is being used to facilitate the loss for the taxpayer.

While legitimate tax shelters do exist, such as depreciation claimed for investment in residential property that houses low-income tenants, a loss is generally not allowed unless it results in an actual loss for the taxpayer [8].

3. Modeling Shelter Risk for Individual Taxpayers

While electronic filing is becoming more popular, the majority of the tax forms submitted by those involved with abusive tax shelters are not filed electronically. Therefore, most are manually transcribed. If all line items for every return were accurately recorded and available electronically, it would make the job of identifying potentially abusive transactions much easier. Unfortunately, only a subset of the line items are actually transcribed and available in electronic form. and the values that are available are sometimes questionable [9].

We began the modeling process by working with an IRS technical advisor to identify the type of behavior the IRS is interested in identifying. The two principal methods for abusively sheltering income from taxes include manufacturing offsetting losses [without any real loss for the taxpayer] and not reporting income. Identifying abusive offsetting losses is considered to be lower hanging fruit, however, because taxpayers are encouraged to actually record their transactions accurately. There is a three year statute of limitations for shelters that are reported, while there is no statute of limitations for income that goes unreported. Additionally, possible fines and penalties are much stronger for unreported income [10]. While MITRE has done some work in the area of identifying unreported income, here we report our work dealing with offsetting losses.

The existing system used by the IRS for targeting compliance issues is constructed by analyzing audit results for a set of randomly selected tax returns. Unfortunately, since truly egregious tax shelters are relatively rare (currently believed to occur in about 1% of the high-income taxpayer population), random selection of audits is unlikely to capture egregious transactions. This explains why some truly egregious shelters may receive a low score. In the future, weighted sampling might be used to improve coverage in this relatively rare portion of the population; e.g. computing the selection probability based on the proportion of total positive income being reported as taxable income. For the majority of high-income taxpayers, this proportion is substantial. As illustrated in figure 3, the mode of the distribution occurs around the 88th percentile. In the mean time, we explored the use of kernel-based techniques as an alternative for initial ranking of tax returns for review by a compliance expert.

Given a database with a few known examples of "abusive" transactions and no other "labels," we pursued constructing a single-class model to measure the similarity of high-income taxpayer relationships to known examples of abuse. We started with a half-dozen examples of abuse provided by the technical advisor, but discovered there was no data connecting the pass-through entities to the high-income taxpayers. We compensated for the lack of training examples by asking the technical advisor to describe the behavior of interest and querying the database to find matching examples. More than 50 variables were being considered for each set of entities connected to a high-income taxpayer. The query output was processed slowly, as there was a significant learning curve for the data miners trying to understand how the subset of transcribed line items from different tax returns related to one another. Note: Because the IRS wants to encourage electronic filing, the IRS does not use untranscribed line items from electronic returns for targeting compliance issues.

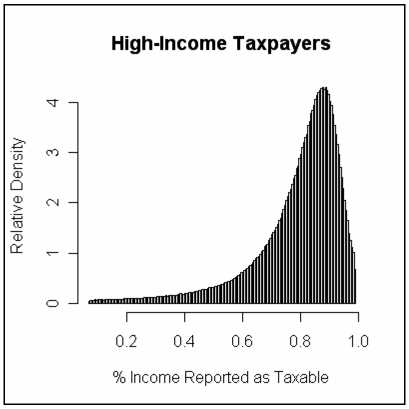


Figure 5: Taxable Income

Our initial query involved searching for one to four high-income taxpayers receiving little income from an initial year partnership and large losses from an initial year S corporation. While the first couple of matches proved we needed to refine the targeting criteria, later examples indicated the existence of multi-million dollar shelters that had been previously undiscovered. An existing compliance project was used to support initiating audits on the discovered shelters.

After obtaining 30 examples, we constructed a single-class model using a Support Vector Machine (SVM) to produce a similarity measure for weighting the loss in question. Training the single-class SVM [11] was described to the IRS domain experts as being similar to how you might train a revenue agent. First, a few positive examples are provided in terms of the features relevant to identifying a potentially abusive transaction. The "trainee" is then asked to evaluate a new set of returns to identify similar behavior. In this case, the "trainee" is also expected to identify the source of the suspicious loss.

Somewhat redundant features were used to provide robustness against transcription errors; e.g. using line items describing short term capital losses from both schedule K-1 of the pass-through entity and the high-income taxpayer's return. Instead of having the single-class SVM produce a TRUE/FALSE class label, however, we used the raw sum of the kernel (similarity) function output to compute a risk metric for ranking; i.e. we used a Gaussian kernel to compute the similarity between each transaction and the optimal prototypes from the known abusive transactions (training data). The "nu" parameter of the SVM was used to allow a small subset of examples to be declared to be outliers, while the "gamma" parameter of the kernel was used to allow selection of the optimal prototypes (support vectors) by favoring less complex models providing the best coverage of the known abusive examples. To find the best hyper-parameter values, a hierarchical grid search was conducted over the range of feasible "nu" and "gamma" values using leave one out cross validation.

It takes only a few minutes to construct the model, and it takes only an hour to assess risk for a year of tax return data. The longest part of the process involves preprocessing the data by deriving features from the line items of the returns and normalizing the feature vectors to unit length (so a $1 million dollar offset of a $1 million dollar income is given the same weight as a $100 million dollar offset of a $100 million dollar income and the resulting weights are then multiplied by the magnitude of the sheltered income to assess overall risk).

This model was successful for identifying and ranking a few specific types of transactions, revealing an estimated $200 million dollars of previously undiscovered shelters. This model was also useful for providing coverage of substantially similar transactions. Nevertheless, while precision for this model was around 90%, the recall was suboptimal. Transactions were being missed due to transcription issues and the use of similar transactions with different types of assets; e.g. foreign currency straddles characterized as "ordinary loss" instead of stock option straddles characterized as "short term capital loss".

Based on feedback from the domain experts, we decided to generalize the targeting strategy by relaxing the targeting criteria to review a smaller, more general set of targeting features; e.g. total positive income, largest gain, largest loss, taxable income, etc. This simplistic model is known as the Shelter Risk Function (SRF). The values associated with a transaction for a high-income taxpayer are compared to an idealized shelter using a Gaussian kernel. The kernel width was selected using a jackknife procedure [13] to identify the value that produces the highest correlation to audit results from a prior tax year.

For our initial evaluation, an idealized shelter is characterized as a single source of income being offset by a single source of loss, resulting in zero taxable income. Again, somewhat redundant features are employed to provide robustness against transcription errors. The similarity of data describing a taxpayer and an idealized shelter is used as a weight for the income that is being sheltered. The results of the risk assessments for this model are then fed to an associative analysis engine to identify groups of related shelter suspects.

4. Modeling Shelter Risk for Groups of Taxpayers

While SRF can provide a reasonable targeting metric for identifying potentially abusive shelter activity, it does not attempt to identify common links between shelters. Some shelters are customized for an individual and are not tightly linked to additional shelters. Other shelters, however, share a common structure and mode of operation, which having been designed once can be sold to different clients over and over by a shelter promoter [14].

Figure 4 illustrates an example of a group of approximately 40 related shelters. On the far left is the promoter that created the tax shelter, and on the far right is a pair of entities that "sell" this tax shelter to taxpayers. The outer ring of the picture is a set of high-income individuals who are sheltering their income. The ring in the middle is a set of partnerships manufacturing abusive tax shelters using straddles. The two entities in the middle are foreign partners, tax indifferent parties claiming the allocation of gain from the straddles. This promotion was used to shelter a truly large amount of income for high-income taxpayers.

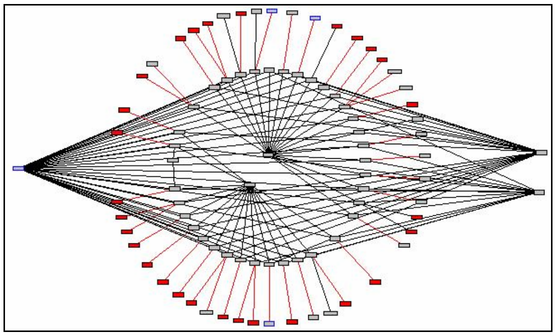


Figure 6: Group of Related Tax Shelters

These “cookie cutter” shelter promotions are particularly interesting to the IRS because they make it necessary to argue only one instance of the shelter in court. If the IRS wins the case, or a few of these cases, it becomes more likely that the other instantiations will not go before the court, and further litigation costs are saved. When it is considered that some abusive shelters are worth tens of millions of dollars or more to their recipients, it is not surprising that they are defended vigorously. Detecting promotions rather than just single tax shelters can thus be highly advantageous for the IRS in terms of reduced cost of enforcement.

We developed the Promoter Risk Function (PRF) to be used in conjunction with SRF with the goal of grouping SRFs and other high income individuals together with businesses and individuals potentially promoting abusive tax shelters. PRF is a custom link analysis application that explores the relationship between groups of SRF suspects and various identifiers of the potential promoter nodes, including their names, addresses and Taxpayer Identification Numbers (TINs). Preparer information is included in the attributes being examined.

Promotions such as the one shown in figure 4 helped to suggest our approach. Each participant in this promotion receives his own shelter from an individual partnership, but those partnerships are in turn all connected to just a handful of promoter entities. Given a subset of this shelter's participants, it is possible to traverse their K-1 links to discover potential promoter entities. By reversing this process—expanding the search outward from the newly discovered promoter suspects—not only will the original suspects be reached, but the other individuals associated with the promotion as well. In this way we can group a set of suspects into various suspected shelter promotions and also discover previously unsuspected shelter participants.

We generally use SRF as the starting point for promotion detection with PRF, although other targeting functions can be used as well. PRF starts with a specific target group (hereafter referred to as suspects), but as described above it also discovers other high income SSNs that are associated with suspected promoter attributes. This is beneficial in that not only does it yield better recall of promotion participants, but it also allows PRF to judge the likelihood of particular groupings of suspects and non-suspects, as discussed below.

5. Pruning Links

The most naive implementations of PRF will not run to completion in a reasonable amount of time, due to combinatorial explosion. Some nodes may have as many as hundreds of thousands of connections to others. Traversing these nodes even once is too much, and caching such results is not feasible with even 4 gigabytes of system memory available. In order to reduce the run time and storage requirements, we implemented some simple pruning heuristics.

Connections between nodes were cut when the number of investors or investments was greater than threshold, unless the link represented more than 10% of the equity for that node. For the investments threshold, Chebyshev's inequality [15] was used with k = sqrt(20) to identify inbound links where the payee has an unusually large number of payers. For the investors threshold, a domain expert defined rule was used to identify outbound links in which the payer has more than 10 payees. This reduced the number of possible links to be analyzed from 24.7 million links to 16.8 million links.

Additionally, we limit depth of search to two hops from the starting points, the input suspects. Unlike the pruning heuristics, this heavily affects the behavior of PRF in both positive and negative ways. The limited search could obscure more complicated shelter promotion schemes in which invariant promoter attributes are always more than two hops away from the suspects. However, there is a hidden advantage here: by limiting the search depth, spurious connections between suspects are encountered less often, helping to limit false positives. A final step was to restructure the database to reduce the number of disk reads required to process the links.

These measures reduced the execution time by orders of magnitude so that the PRF tools can complete a search of a given tax year in approximately three to four hours on our hardware (a 2.8 gigahertz Intel Xeon processor with 4 gigabytes of memory).

6. Filtering and Grouping

After the database has been searched for links, we take additional steps to filter and group the data. The most important filtering stage is to threshold potential promoter identifiers based on their level of support in the input group. Those identifiers linked to smaller numbers of suspects are less likely to be part of a promotion. In the degenerate case, when there is only a single suspect linked to an identifier, the identifier has no resolving power in the discovery of a shelter promotion.

If the support threshold is met for a given potential promoter, we generate the odds ratio of the number of suspects to non-suspects associated with the potential promoter, compared to the ratio of the total number of other suspects to the total number of other non-suspects in the population. The greater the odds ratio, the less likely a group with the same number of suspects and non-suspects would be generated from a random sampling of the population. While a p-value from a Chi-square or Fisher test [16] can be used to evaluate the hypothesis that P (Suspect = True | Link to

X) ≤ P(Suspect = True | No Link to X), a p-value is not so useful for measuring the degree of association between the conditions "Suspect = True" and "Link to X".

Consider the contrast provided by contingency tables 1 and 2. The one-sided Fisher test p-value for table 1 is $5.7*10^{-183}$, while the p-value for table 2 is $0.1*10^{-183}$. Yet the odds ratio for table 1 is about 1868, while the odds ratio for table 2 is only 25 (a smaller p-value indicates the null hypothesis is less likely to be true, while a larger odds ratio indicates a stronger association between the two factors). Intuitively, the odds ratio result makes more sense because 95% of the returns associated with preparer A are associated with shelter suspects, while less than 20% of the returns associated with preparer B are associated with shelter suspects.

		Shelter Suspect?	
		Yes	No
Prepared by A?	Yes	95	5
	No	19,905	1,979,995

Table 1: Contingency Table for Preparer A

		Shelter Suspect?	
		Yes	No
Prepared by B?	Yes	197	803
	No	19,803	1,979,197

Table 2: Contingency Table for Preparer B

A final grouping stage is run to associate promoter identifiers with other promoter identifiers by thresholding on the Jaccard similarity [17] between the identifiers' suspect groups. When the similarity between the suspects is sufficiently high, we consider the two groups to be part of the same potential promotion and to form part of the same metagroup.

7. Results

With typical threshold parameter settings, PRF finds on the order of 500 metagroups of potential promoters and SSNs for every year, which combine to total a few billion dollars of sheltered income. Around 50% of this total is associated with the top 20 metagroups, as ranked by a combination of the lower bound of a confidence interval for the odds ratio [16] of the metagroup and the amount of income suspected of being sheltered.

When comparing these top 20 metagroups to a list of known shelter participants from the IRS, there is a substantial overlap between the taxpayers believed to have participated in an abusive tax shelter. Examining the PRF results more closely, there appear to be several advantages PRF can provide in addition to its goal of appropriately grouping SRF suspects with potential promoters.

First, it is able to use the initial suspect list to discover other high income individuals that, while not passing the SRF threshold, do appear to be participating in abusive shelters. This is borne out in the high proportion of non-SRFs in the PRF output that overlap the IRS's list of known shelter participants.

Beyond that, PRF is also able to automatically discover shelter participants unknown to the IRS. PRF also maintains the links associating nodes in its output clusters, reducing the effort required to verify whether a suspected participant in a promotion really is taking the shelter.

Perhaps the biggest advantage to PRF is its ease of use and relative efficiency. One of the largest promotions for tax year 2001 required roughly two weeks of work by multiple IRS agents to trace out. PRF is able to do much of that work in just a few hours, with no human intervention. Further it discovers participants that the IRS may not have otherwise found. Running PRF as soon as the data for each tax year is received has the potential to find the most egregious tax shelters efficiently and quickly, with minimal auditor labor.

8. Conclusions

Recent efforts to combat abusive tax shelters have met with some success [18]. While MITRE is certainly not solely responsible for this outcome, we did play a role in helping to identify abusive shelters. After review of the output by domain experts, audits for selected cases resulted in substantial assessments for additional tax collection by the IRS. Nevertheless, more work remains to be done, such as adjusting the risk values by accounting for differences in tax rates; e.g. the most widely paid tax rate for capital gains is 15%, while the tax rate for ordinary income is often more than 30%. Further work is also needed in assessing risk for unreported income, as well as understanding how abusive promotions change over time.

9. References

[1] IRS Publication 3744, "IRS Strategic Plan: 2005 - 2009", Rev 6-2004, p.18, http://www.irs.gov/pub/irs-utl/strategic_plan_05-09.pdf.

[2] IRS Form 1041 Schedule K-1,"Beneficiary's Share of Income, Deductions, Credits, etc." 2003, ftp://ftp.irs.gov/pub/irs-03/f1041sk1.pdf.

[3] IRS Form 1065 Schedule K-1,"Partner's Share of Income, Deductions, Credits, etc." 2003, ftp://ftp.irs.gov/pub/irs-03/f1065sk1.pdf.

[4] IRS Form 1120S Schedule K-1,"Shareholder's Share of Income, Deductions, Credits, etc." 2003, ftp://ftp.irs.gov/pub/irs-03/f1120ssk.pdf.

[5] United States Code Title 26 Section 1362,"Subchapter S - Tax Treatment of S Corporations and their Shareholders", 1982,http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi

?dbname=browse_usc&docid=Cite:+26USC1362.

[6] GAO Report GAO-02-618T, "Enhanced Efforts to Combat Abusive Tax Schemes - Challenges Remain", Apr 2002, p.10, http://www.gao.gov/new.items/d02618t.pdf.

[7] IRS Notice 2000-44, "Inflated Partnership Basis Transactions (Son of BOSS)", Sep 2000, http://www.irs.gov/pub/irs-utl/notice_2000-44.pdf.

[8] IRS Tax Topic 454, "Tax Shelters", http://www.irs.gov/taxtopics/tc454.html.

[9] Government Accounting Office Report GAO-04-1040, "IRS Should Take Steps to Improve the Accuracy of Schedule K-1 Data", Sep 2004, pp.3-4, http://www.gao.gov/new.items/d041040.pdf.

[10] United States Code Title 26 Section 6663,

"Imposition of Fraud Penalty", 1989, http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi

?dbname=browse_usc&docid=Cite:+26USC6663.

[11] B. Scholkopf, J. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson, "Estimating the Support of a High-Dimensional Distribution", Neural Computation, 13, Jul 2001, pp.1443-1471.

[12] IRS Notice 2002-65, "Passthrough Entity Straddle Shelters", Sep 2002, http://www.irs.gov/pub/irs-utl/notice_2002_65_(l21).pdf.

[13] B. Efron and R.J. Tibshirani, "An Introduction to the Bootstrap", May 1994, pp.141-150.

[14] Senate Report 109-54, "The Role of Professional Firms in the U.S. Tax Shelter Industry", Apr 2005, http://frwebgate.access.gpo.gov/cgi-bin/useftp.cgi ?IPaddress=162.140.64.88&filename=sr054.pdf

&directory=/diskb/wais/data/109_cong_reports.

[15] S. Ghahramani, "Fundamentals of Probability", Prentice Hall, 2nd ed, Sep 1999, pp.437-441.

[16] A. Agresti, "Categorical Data Analysis", Wiley-Interscience, 2nd ed, Jul 2002, pp.91-101.

[17] D.J. Hand, H. Mannila, and P. Smyth, "Principles of Data Mining", The MIT Press, Aug 2001, p.37.

[18] IRS Newswire IR-2004-87, "Strong Response to 'Son of BOSS' Settlement Initiative", Jul 2004, http://www.irs.gov/newsroom/article/0,,id=124937,00.html

10. Acknowledgements

A number of people provided critical support for this work. The IRS compliance experts included: Mike Whalin, Susan Kerrick and Scott Emerson. The IRS research experts included: Mark Mazur, Larry May, John Davidson, Kay Wolman, Jim Needham, Maury Harwood and Christine Smith. Our colleagues at MITRE included: Anne Cady, Dan Venese, Lowell Rosen and Sarah Piekut. This work was performed under IRS contract number TIRNO-99-D-00005.

Data-driven Modeling of Acute Toxicity of Pesticide Residues as Alternative Tool within Official Registration, Evaluation and Authorization Procedures

Frank Lemke[1], Emilio Benfenati[2], Johann-Adolf Mueller[1]
[1]*KnowledgeMiner Software, Berlin, Germany*
[2]*Istituto di Ricerche Farmacologiche "Mario Negri", Milan, Italy*
frank[ατ]knowledgeminer.net
benfenati[ατ]marionegri.it
jamueller[ατ]knowledgeminer.net

Abstract

This paper outlines and implements a concept for developing alternative tools for toxicity modeling and prediction of chemical compounds to be used for evaluation and authorization purposes of public regulatory bodies to help minimizing animal tests, costs, and time associated with registration and risk assessment processes. Starting from a systems theoretical analysis we address and introduce concepts of multileveled self-organization for high-dimensional modeling, model validation, model combining, and decision support within the frame of a knowledge discovery from noisy data.

1. The problem of ecotoxicity

Besides the economical importance of the chemical industry as Europe's third largest manufacturing industry, it is also true that certain chemicals have caused serious damage to human health resulting in suffering and premature death and to the environment. The incidence of some diseases, e.g. testicular cancer in young men and allergies, has increased significantly over the last decades. While the underlying reasons for this have not yet been identified, there is justified concern that certain chemicals play a causative role for allergies.

The global production of chemicals has increased from 1 million tons in 1930 to 400 million tons today. There are about 100.000 different substances registered in the EU market of which 10.000 are marketed in volumes of more than 10 tons, and a further 20.000 are marketed at 1-10 tons per year. The present system for general industrial chemicals distinguishes between "existing substances" i.e. all chemicals declared to be on the market in September 1981, and "new substances" i.e. those placed on the market since that date. There are some 3000 new substances. Testing and assessing their risks to human health and the environment according to the EC Directive 67/548 are required before marketing in volumes above 10 kg per year. For higher volumes more in-depth testing, focusing on long-term and chronic effects, has to be provided [1]. In contrast, existing substances amount to more than 99% of the total volume of all substances on the market, but they are not subject to the same testing requirements. Some of them have never been tested at all. The number of existing substances reported in 1981 was 100.106, the current number of existing substances marketed in volumes above 1 ton is estimated at 30.000. In result, there is a general lack of knowledge about the properties and the uses of existing substances. The risk assessment process is slow and resource-intensive and does not allow the system to work efficiently and effectively [1].

To address these problems and to achieve the overriding goal of sustainable development one political objective formulated by the European Commission in its "White Paper on the Strategy for a Future Chemicals Policy" [1] is the implementation of the so-called REACH system (*R*egistration, *E*valuation and *A*uthorization of *Ch*emicals). Some more important objectives of the REACH framework are the protection of human health and the environment, an increased overall registration transparency, integration with international efforts, and the promotion of non-animal testing methods.

Based on World Bank estimates and a number of prudent assumptions, diseases caused by chemicals are assumed to account for some 1% of the overall burden of all types of disease in the EU. Assuming a 10% reduction in these diseases as a result of REACH would result in a 0.1% reduction in the overall burden of disease in the EU. This would be equivalent to around 4.500 deaths being avoided every year [2].

Due to lack of data it is not possible to get a quantitative idea of the impacts on the environment. All in all, however, it is expected that REACH will contribute to reduced pollution of air, water, and soil as well as to reduced pressure on biodiversity and to reduced effects from endocrine disrupting chemicals [2].

According to a study of the University of Leicester, UK, one cost for implementing REACH would be an additional need of about 12 million animals for testing purposes. Because of this costs and the very long time it would take to run animal tests for all chemicals to be assessed (> 30.000), alternative, standardized, validated and accepted, by both industry and regulatory bodies, non-animal test methods are required. Current estimates expect that such alternative methods would save the lives of at least 2 million animals [3].

A current and promising way in that direction is building mathematical models – QSARs, Quantitative Structure-Activity Relationship models - based on already existing test data that describe and predict the impact of a given dose or concentration of a chemical compound (pollutant) on the health of a population of a certain biological species by the chemical's molecular properties. Typical parameters that are used in QSAR for expressing the chemical's impact on the population's health are the lethal dose LD_{50} or the lethal concentration LC_{50}. LC_{50}, for example, specifies the experienced concentration of a chemical compound where 50% of the population died within a given time, for example within a period of 96 hours ($LC_{50/96h}$), after introduction of the chemical into the system.

2. The information base

Besides the ethical, cost, and time considerations of running traditional bioassays to evaluate the ecotoxic effects of a chemical, there are also methodological problems. Ecotoxicological systems are complex, ill-defined systems, which are characterized by [4]:

1. Inadequate a priori information about the system. Creating models for predicting toxic or other negative effects on the environment and human health is a highly interdisciplinary challenge. Scientists from chemistry, toxicology, biology, systems theory, information technology and machine learning, but also, not to forget, end users from industry and public, regulatory bodies have to work together for finding a real working solution. There is no domain knowledge available, from any single domain, that would suffice to solve the problem by theory-driven approaches.

2. Large number of potential, often immeasurable or simply unknown variables. A few hundred to a few thousand input variables are not uncommon in toxicity QSAR modeling.

3. Noisy and few data samples. Reliable experimental toxicity data derived from past bioassays are rather rarely available and to obtain. Some tens to a few hundred data samples are common in toxicity QSAR modeling, though.

4. Fuzzy objects. Experimental toxicity data are result of animal tests. Depending on the species used in an assay its inherent bio-variability can be quite high and can vary very much from species to species.

The economical, ethical, and methodological problems resulting from applying traditional bioassay and theory based methods but also dedicated expert systems [5] suggest and demand using a data–driven approach for finding an alternative tool for the evaluation and authorization of the huge amount of chemicals on the market.

2.1. Systems theoretical view

Generally, real-world systems are time-variant nonlinear dynamic systems [6] usually described by systems of nonlinear difference equations. For acute toxicity modeling this system can be considered time-invariant due to the intentionally short-term effect of the pollutant.

A corresponding dynamic model of the ecotoxicological system under research is shown in figure 1.

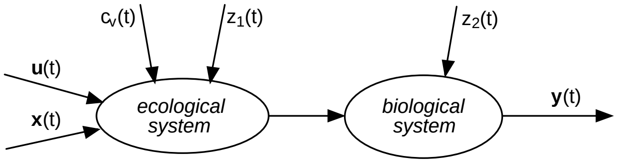


Figure 1. Dynamic model of an aquatic ecotoxicological system

Where
$\mathbf{x}(t)$ – state vector of the ecological system at time;
$\mathbf{u}(t)$ – vector of external variables at time t;
$c_v(t)$ – concentration of the pollutant v at time t, with

$$c_v(t) = \begin{cases} c_0 & t = t_0 \\ 0 & \text{else} \end{cases}$$

, and c_0 as the concentration of the test compound v in mg/l, for example;
$z_1(t)$, $z_2(t)$ – external disturbances to the system at time t;
$\mathbf{y}(t)$ – output vector of dimension p describing the health of the population at time t,

$\mathbf{y}(t)=[y_1(t), y_2(t),.., y_m(t), .., y_p(t)]^T$;
$y_m(t)$ – the population's cumulated mortality rate at time t (see also fig. 3).

When running animal test series in laboratories according to some standard protocols to obtain experimental toxicity values the external variables and the state variables of the ecotoxicological system are not observed or not observable, usually, and therefore they are considered constant so that for modeling the ecotoxicological system transforms into a nonlinear static system [7] (fig. 2):

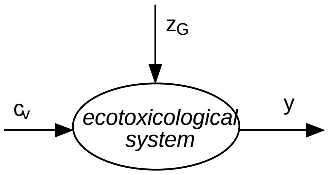


Figure 2. Reduced model of the static system with noise z_G

Additional noise z_3 is introduced to the static system by the missing information of external and state variables that now transforms to noise. Also, the testing procedure itself adds some noise z_4 so that the static system's noise finally is $z_S = h_1(z_G, z_3, z_4)$, and the modeling task of the ecotoxicological system reduces to approximating the dependence of the experienced mortality rate y from the pollutant's concentration c_v: $y = f_1(c_v, z_S)$.

If an animal test series is repeated several times under the same test conditions and standard protocols, for a given concentration $c_{i,v}$ of a chemical test compound v, multiple experienced mortality rate values are observed (fig. 3). This means, for $c_{i,v}$ = const., the interval of the observed mortality rate values can be seen as a direct expression of the static system's noise z_S.

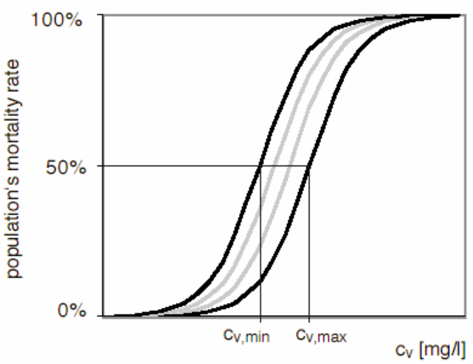


Figure 3. Variation of LC50 resulting from a number of comparable tests

For the reverse case of measuring the concentration c_v for a constant mortality rate y_j = const. the problem transforms to $c_v = f_2(y_j, z_S)$ (fig. 3).

For a mortality rate y_j = 50%, c_v is called the experienced lethal concentration LC_{50} for a pollutant v, which is finally used as the output variable in toxicity QSAR modeling. With a commonly observed rate $\frac{c_{v,max}}{c_{v,min}} \approx 4$ for a single compound v this output variable can be seen as highly noisy.

The initial task of modeling the observed mortality rate y from a pollutant's concentration c_v now shifts to finding a description of the dependence of a pollutant's lethal concentration LC_{50} for a specific species from a chemical pollutant's molecular structure $\mathbf{s}_v$ (fig. 4) with $z_M = h_2(z_S)$ and $LC_{50} = f_3(\mathbf{s}_v, z_M)$.

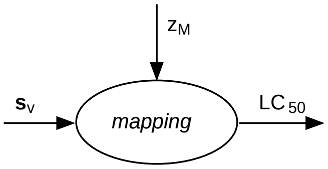


Figure 4. The toxicity QSAR modeling problem

This finally means not to model the object itself – the ecotoxicological system – but one of its inputs – the external disturbance c_v and the initial system's input-output relation is just mapped by a single pair of observations (LC_{50}, y).

A next problem is how to express the structure $\mathbf{s}_v$ of the chemical v. Commonly, it is a complex chemical object, but for building a mathematical model that describes the dependence of the toxicity from its chemical structure $\mathbf{s}_v$, a formal transformation into a set of numerical properties – descriptors $\mathbf{d}_v$ - is required. This transformation is based on chemical and/or biological domain knowledge implemented in some software with $\mathbf{d}_v = f_4(\mathbf{s}_v, z_T)$ (fig. 5).

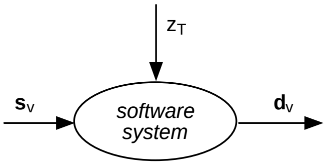


Figure 5. Model of the chemical structure to molecular descriptor transformation

In the chemical domain, for example, the input $\mathbf{s}_v$ of the software system can be a 2-dimensional or a 3-dimensional drawing of the chemical structure, but also other expressions may be possible. Output of the system is a certain set of molecular descriptors $\mathbf{d}_v$ that depends on the software package used and the theoretical model implemented. Applying different software tools provides different sets of descriptors that may intersect to some extent but may not necessarily have identical values though. Also, the interpretational power of descriptors can be low or difficult when they loose chemical or biochemical meaning.

The process of descriptor calculation also adds noise. Not only software bugs or manual failures may introduce noise, more important for introduction of uncertainty should be the interpretational clearance of domain knowledge for properly formalizing an appropriate set of molecular descriptors, different starting condition assumptions (conformation) for descriptor calculation, or several different optimization options. Not always is their chemical meaning very strong or theoretically accounted.

The final, simplified nonlinear static model used in QSAR modeling to describe acute toxicity is shown in figure 6:

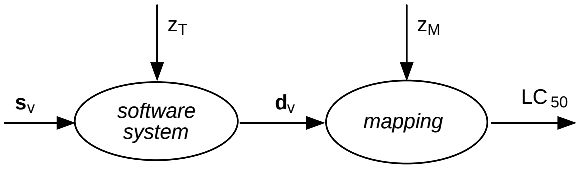


Figure 6. Simplified model for describing acute toxicity

With
$LC_{50} = f_5(f_4(s_V, z_T), z_M) = f(s_V, z_T, z_M)$,
and
LC_{50} – experienced lethal concentration for a certain species and chemical compound,
$\mathbf{s}_V$ – the structure of the tested chemical compound in the chemical domain,
z_T – noise of the chemical structure to molecular descriptor transformation process,
z_M – noise transformed from the ecotoxicological test system,
$\mathbf{d}_V$ - vector of numerical molecular descriptors of the test compound

The external disturbance z_T which adds noise to descriptor input space used for modeling can be reduced by fixing bugs and manual failures and by finding a most consistent chemical structure-to-descriptor transformation – although it is not clear a priori which transformation or optimization will add and which will reduce noise. The disturbance z_M, which finally results from the experimental animal tests, in contrast, adds noise to the output LC_{50} and is a given fact that cannot be changed afterwards.

2.2. The data sets

Biological data are affected by factors relative to the biological system itself and to factors dependent on the investigation technique used. While natural variability cannot be eliminated, and is part of the real world, many attempts have been done to reduce the influence of the technique used to study the biological system, through the introduction of standardized procedures. Commonly, the term variability is used in relation to the natural factors, while uncertainty is used in the case of factors related to the technique to study the biological phenomenon. In our case, we used only data on pesticide ecotoxicity originating from experiments, which have been conducted according to official guidelines. In particular, Dr. Brian Montague from the US Environmental Protection Agency, Washington, DC, provided the data for this work. In many cases several different values for the same compound was reported, resulting from different experiments conducted all according to the official guidelines. We defined some criteria for the selection of appropriate values, in order to use experiments with a higher quality and a lower variability [8]. Furthermore, we checked the values with other databases, in order to increase their reliability. We studied five different toxicological endpoints, and the number of compounds was less than 300 in the most favorable case (toxicity towards rainbow trout) to about 100 in the case of bee toxicity. The limited number of examples is, indeed, a common problem for this type of study, mainly – like in our case on pesticides - when a heterogeneous set of compounds is used, referring to many different kinds of bio-mechanisms responsible for the observed toxicity phenomenon.

The issue of the prediction of ecotoxicity of pesticides was considered within the EC funded project DEMETRA [9]. The major objective of this project is to produce software for toxicity prediction of pesticides and related compounds (such as metabolites), directly and immediately useful for evaluation of pesticides and related compounds within the Dossier preparation for pesticide registration. The software aims to be used by end users such as regulatory bodies, industries, nongovernmental organizations, and researchers. It will allow processing of chemicals, one by one, for prediction of toxicity for pesticides and related compounds. It will also support regulatory evaluators to assess data submitted in approvals applications.

Compared to the general target of the prediction of toxicity of industrial chemicals, as discussed in section 1, the target of DEMETRA is more focused to pesticides, and thus somehow more difficult, because pesticides are typically very active compounds, complex on a chemical point of view (many functional groups are present, often several of them within the same compound) and on a toxicological point of view (for the occurrence of many toxic mode of action caused by the compounds). Furthermore, pesticides are limited, and the number of data available is small.

To describe the chemical nature of the compounds we used several software tools, such as DRAGON, CODESSA, Pallas, Cache. As a result, thousands of molecular descriptors are available for each chemical compound.

3. Data mining approach

The general problem to solve in this work is building mathematical models based on existing test data that describe and predict the short-term, acute impact of a given dose or concentration of a chemical compound (pollutant) on the health of a population of a certain biological species. There are rather few experimental toxicity test data available, which are also considerable noisy (section 2.1), while a very large number of potential input variables are describing the properties of the chemical compounds. This results in dealing with a high-dimensional modeling problem. Furthermore, there is only very limited domain knowledge that could be used for modeling purposes so it calls for tools that perform a knowledge extraction from data.

3.1. The problem of toxicity prediction

Apparently, inductive modeling of real life systems implies dealing with very noisy data. Data sets generally are not perfect reflections of the world. The measuring process necessarily captures uncertainty, distortion and noise. Noise is not errors that can infect data but is part of the world. Therefore, a modeling tool, but also results and decisions, must deal with the noise in the data. For a small level of noise dispersion, all regression-based methods using some internal criterion can be applied: self-organizing Statistical Learning Networks (also known as Group Method of Data Handling; GMDH [4, 10, 11]) with internal selection criteria, statistical methods, or Neural Networks. For considerably noisy data – which always includes small data samples – GMDH or other algorithms based on external criteria are preferable. For a high level of noise dispersion, i.e., processes that show a highly random or chaotic behavior, finally, nonparametric algorithms of clustering, Analog Complexing pattern recognition, or fuzzy modeling should be applied [4, 12] to satisfy Stafford Beer's adequateness law [13]. This implies, of course, that with increasing noise in the data the model results and their descriptive language become fuzzier and more qualitative too.

Practical data mining application has to handle mountains of data, i.e. tables with high dimension. Besides the known theoretical dimensionality problem there is also a dimension limit of all known tools regarding computing time and physical memory. Therefore a step of high priority is the objective choice of essential variables - state space reduction. In many fields, such as toxicology, there are only a small number of observations but many observed or calculated variables, which is the reason for uncertain results.

There is a broad spectrum of possible algorithms to be used, because it is not possible to exactly define the characteristics of the object under study in advance. Therefore, it is helpful to try several modeling algorithms, first, and then decide which algorithms suit the given type of object best or which most appropriately combine the results of different modeling runs in a hybrid model. Due to the large noise level in toxicity modeling Descriptive Power [14] (see also section 4) might also be part of the model evaluation procedure, because models that can be interpreted from a theoretical viewpoint can be judged using domain knowledge.

3.2. High-dimensional modeling

Reducing the dimensionality requires enhancing the relationships that are really present in the data. This means that a data set has to be representative. This requirement, however, is most difficult to proof in practice. Secondly, a concentration of samples has to enhance the whole information about between-variable relationships but also the variability of individual variables. In toxicity QSAR modeling, therefore, the focus is on state space reduction.

A rule for data driven approaches with parametric models is that the number of unknown model parameters (and connected to this the number of system variables) must be smaller than the number of observations (samples). On the other hand, the number of observations cannot be extended infinitely, because like many economical and ecological systems, for example, toxicity QSAR modeling is characterized by a strongly restricted set of available observations. If the number of system variables in a data set is larger than the number of observations, the modeling task is called an under-determined task. Such under-determined tasks can be solved by means of inductive selection procedures like self-organizing GMDH networks.

Complex systems require measuring many system variables since the necessary dimension of the state space, which the system trajectory is completely described in without redundancy, is commonly not known. More variables presumably carry more information about how the system behaves, but having too many input variables, soon brings computational difficulties and may defect any modeling method because of a combinatorial explosion of the resulting calculations. If the number of system variables is large traditional modeling methods quickly become ill behaved and computational unmanageable.

Another more important problem that high dimensionality presents for data mining tools is that as the number of variables increases, the size (multidimensional volume) of the state space increases too. This would require using more data samples to fill the space to any particular density. In low-density spaces there is higher probability of overfitting a model than in more populous state spaces. To create a representative model a data mining tool ideally needs at least one sample for each meaningful system state present in the state space. Therefore, the number of observations required can become very quickly impractical.

In the literature are several methods for state space reduction proposed, such as:

- Compression by means of Principal Component Analysis or Factor Analysis;
- Removing by filter approaches such as forward, backward and stepwise regression but also by methods based on Bayesian decision or information theory (entropy analysis), rough sets, correlation, least effective element.

Our experience has shown several problems from applying these tools, for instance, in all cases there are many practical problems in discovering high-dimensional nonlinear correlations.

In our work we introduced and used a wrapper approach, where the selection of relevant variables is evaluated by a data mining algorithm, i.e., by the quality of results or the appropriateness of a variable to contribute to solving the given modeling task. In this case, variable selection is based on the so far reached model quality in the data mining process, i.e., we have an iterative procedure. This approach, of course, is very computational intensive.

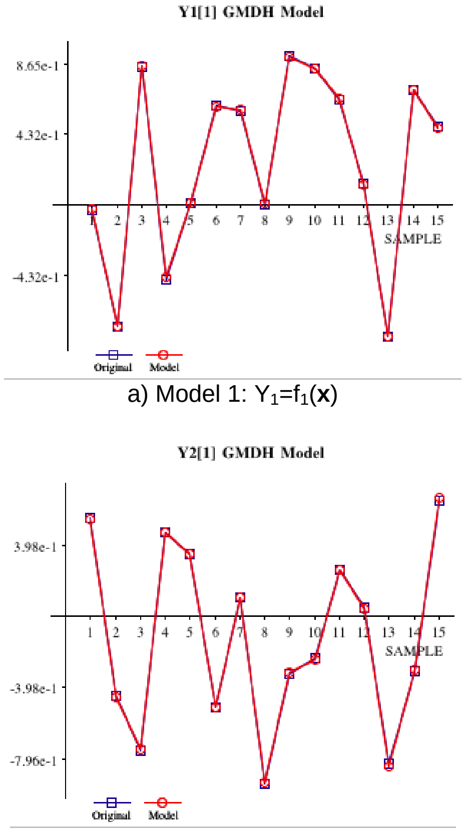
Our new approach to high-dimensional state space modeling we have been developing is based on multileveled self-organization [4]. The basic idea here is dividing high-dimensional modeling problems into smaller, more manageable problems by creating a new self-organizing network level composed of active neurons, where an active neuron is represented by an inductive learning algorithm in turn (lower levels of self-organization) applied to disjunctive data sets. The objective of this approach is based on the principle of regularization of ill-posed tasks, especially on the requirement of defining the actual task of modeling a priori to allow the algorithm selecting a set of correspondingly best models. In the context of a knowledge discovery from databases, however, this implies using this principle in every stage of the knowledge extraction process – data pre-selection, pre-processing including dimension reduction, modeling (data mining), and model evaluation – consistently. The proposed approach of multileveled self-organization integrates pre-processing, modeling, and model evaluation into a single, automatically running process and it therefore allows for directly building reliable models from high-dimensional data sets (up to 30.000 variables), objectively. The external information necessary to run the new level of self-organization is provided by the corresponding algorithm's noise sensitivity characteristic as explained in [14, 15] (see also section 4).

The inductive learning algorithm we used in the network's active neurons is self-organizing GMDH networks as described in more detail in [4]. In this work a prototype of multileveled self-organization implemented in the "KnowledgeMiner" software tool was used [16].

4. Model validation

A key problem in data mining and knowledge discovery from data is final evaluation of generated models. This evaluation process is an important condition for application of models obtained by data mining. From data mining, only, it is impossible to decide whether the estimated model can reflect the causal relationship between input and output, adequately, or if it's just a stochastic model with non-causal correlations. Model evaluation needs - in addition to a properly working noise filtering for avoiding overfitting the learning data - some new, external information to justify a model's quality, i.e., both its predictive and descriptive power.

Let's have a look at this example [15]: Based on a data set of 2 outputs, 9 potential inputs, and N=15 samples two nonlinear regression models $Y_1=f_1(\mathbf{x})$ and $Y_2=f_2(\mathbf{x})$ were created by a self-organizing data mining algorithm (fig.7) [16].



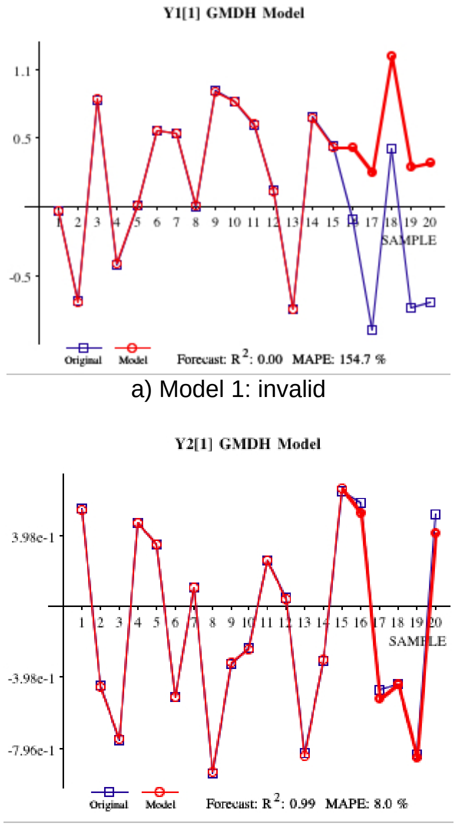
a) Model 1: $Y_1=f_1(\mathbf{x})$

b) Model 2: $Y_2=f_2(\mathbf{x})$

Figure 7. Model graph of two models

For model 1, a Coefficient of Determination (R^2) of 0.9998 and a cross-validated Prediction Error Sum of Squares (*PESS*) of 0.0005 is reported, while model 2 shows a R^2 of 0.9997 and a *PESS* of 0.0006. Concluding from these or any other common model quality or error criteria and from the graphs of figure 7 there is no reason, apparently, to not classify both models as "true" models that reflect a causal relation between output and input. Also, taking into account that the models were created by an inductive, self-organizing model synthesis that implements a powerful active noise filtering during modeling, already [4], it underlines the above assumption.

However, the person who created the data set for this example states that only one model actually describes a causal relationship while the other model simply reflects some stochastic correlations, because output and inputs are completely independent one another (random numbers). Even with this information given - which is usually not the case for real-world data - the modeler cannot decide from the available information which of the two models is the true model. Only applying the models on some new data (which adds new, external information) will turn out the true model (fig. 8):



a) Model 1: invalid

b) Model 2: valid

Figure 8. Prediction of the two models

This example clearly shows that any "closeness-of-fit" measure does not suffice to evaluate a model's predictive and descriptive power, finally. Recent research has shown that model evaluation requires a two-stage validation approach (at least):

1. Level

Noise filtering (hypothesis testing) to avoid overfitting the learning data based on external information not used for creating a model candidate (hypothesis) as an integrated part of the "Learning" process. A corresponding tool that we have been using for a long time successfully is leave-one-out cross-validation, expressed by the PESS criterion.

2. Level

A characteristic that describes the noise filtering behavior of the "Learning" process to justify model quality based on other external information not used in the first validation level yet. This characteristic can be obtained by running a Monte Carlo simulation of a corresponding data mining algorithm many times, so that it expresses a kind of new, independent "common knowledge" that any model can be and must be adjusted with [14].

Figure 9 shows a detail of the noise sensitivity characteristic for linear GMDH Network models implemented in [16]. For nonlinear models a different characteristic was obtained and implemented.

The objective of a second level validation is

(1) That noise filtering implemented in level 1 is very likely to not being an ideal noise filter and thus not working properly in any case (see example above and fig. 9) and

(2) To get a new model quality measure that is adjusted by the noise filtering power of the algorithm.

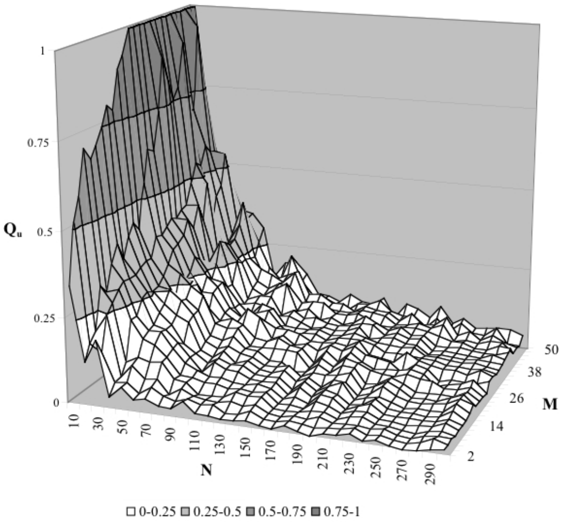


Figure 9. Noise sensitivity characteristic
M: number of potential inputs
N: number of samples
Q_u: virtual quality of a model
Q_u=1: noise filtering does not work at all
Q_u=0: ideal filtering

The noise sensitivity characteristic expresses a virtual model quality Q_u that can be obtained when using a data set of M potential inputs of N random samples. It is virtual model quality, because, by definition, there is not any causal relationship between stochastic variables (true model quality $Q = 0$, by definition [14]), but there are actually models of quality $Q > 0$, which, when using random samples, just reflect stochastic correlations. By implementing an algorithm's noise sensitivity characteristic into a data mining tool it is possible for any given number of potential inputs M and number of samples N to calculate a threshold quality $Q_u=f(N, M)$ that any model's quality Q must exceed to be stated valid in that it describes some relevant relationship between input and output. Otherwise, a model of quality $Q \le Q_u$ is assumed invalid, since its quality Q can also be reached when simply using independent variables, which means that this model does not differ from a model of just stochastic correlations.

In addition to deciding if a model appears being valid or not, the noise sensitivity characteristic is also a tool for quantifying to which extent the data is described by a causal relationship between input and output. This introduces a new, noise filtering and model complexity adjusted model quality measure: Descriptive Power (*DP*), which is defined as:

$$DP=\begin{cases}0 & Q\le Q_u(N,L)\\ \dfrac{Q-Q_u(N,L)}{1-Q_u(N,L)} & Q>Q_u(N,L),Q_u(N,L)<1\end{cases}$$

Here, Q is the measured quality of the evaluated model and $Q_u(N, L)$ is the reference quality calculated from the number of samples N the model was created on and from the number of input variables L the model is actually composed of (selected relevant inputs in the model), with $L \le M$. This means that the Descriptive Power measure excludes any virtual quality that may exist and that it directly allows for model complexity. For example, two models M_1 and M_2 show the same quality $Q = Q_1 = Q_2$, but M_1 uses more relevant inputs than M_2 to reach that quality Q, so, with $L_1 > L_2$, the Descriptive Power of M_2 is higher than that of M_1.

Back to the example above, the implementation of the algorithm's noise sensitivity characteristic in the data mining software [16] now provides additional information in the model report for the two models (fig. 10).

The model cannot be validated, because noise filtering does not work on this insufficiently sized information basis. The obtained model accuracy can also be reached when just using random numbers as input data.
Increase the number of samples to about 75 and/or decrease the number of potential input variables to below 3.
a.) Model evaluation of model Y_1=f_1(**x**)

The model appears to reflect a valid relationship. It describes 98 % of the data.
For the chosen model type, modeling is based on a VERY POORLY sized information basis, which may result in a not properly working noise filtering. To improve the noise filtering capability of the algorithm and thus improving model reliability, it is highly recommended to decrease the variables/sample ratio. If possible increase the Number of Samples: to above 131 and/or decrease the Number of Variables: to below 3.
b.) Model evaluation of model Y_2=f_2(**x**) stating a Descriptive Power of 98%

Figure 10. Reported evaluation results of the two models

The implemented two stage model validation approach now allows, for the first time, to get on the fly an active decision support in model evaluation based on the model's descriptive power calculated on the learning data, only, for minimizing the risk of false interpreting models and using invalid models that just reflect some non-causal correlation.

5. Results

The results shown here were obtained within the DEMETRA EU research project [9].

Based on five data sets - D_1 (Trout), D_2 (Daphnia), D_3 (Oral Quail), D_4 (Dietary Quail), D_5 (Bee), - we first created many individual regression and classification models (> 500) using different modeling and data mining algorithms like Partial Least Squares, different types of Neural Networks, fuzzy modeling, and multileveled self-organization as described in sections 3 and 4.

From this pool of individual models we then created a hybrid model for each data set by combining corresponding individual models.

Since the focus of public regulatory bodies is on regression models, we report results from these models here, only.

5.1. Individual models

Based on the five project data sets a large set of individual QSAR models were created by different project partners using different data mining algorithms. To allow comparison and combination of these models three strict preconditions were defined:

1. The official data sets produced within the DEMETRA project have to be used for modeling, only.
2. Although some of the data sets have rather few compounds, only (~100), each data set D_i was randomly subdivided by a 6:1 split into a learning subset $D_{i,A}$ (or $D_{i,A}$ and $D_{i,B}$) and an out-of-sample test data subset $D_{i,C}$, with $N_{A,B} + N_C = N$. The data in the test subset was never to be used for modeling at all, but was hold out for validating all created individual models on this new data.
3. For comparison purposes, for every model the Coefficient of Determination R^2 calculated on both learning and test data subsets had to be provided:

$$R^2 = 1 - \delta^2, \delta^2 = \frac{\sum_{i \in N} (y_i - \hat{y}_i)^2}{\sum_{i \in N} (y_i - \bar{y})^2} \leq 1,$$

where y_i, $\hat{y}_i$, and $\bar{y}$ are the true, estimated, and mean values of the output variable, respectively. Additional model error or model quality criteria like MSE, MAPE, or Descriptive Power can be calculated and used, too.

The results of the five best individual QSAR models for the trout data set are listed in table 1. Some QSAR models were created using 2-dimensional (2D) molecular descriptors (inputs), only, others were built on 3-dimensional (3D) or 2D and 3D descriptors.

Table 1. Five best models for the data set D_1 – Trout - with respect to $R^2_{A,B,C}$

$R^2_{A,B,C}$	$Q^2_{A,B}$	R^2_C	*m*	*model type*	*DM-method*
0.67	0.69	0.59	10	explicit linear model	multileveled self-organization
0.66	0.66	0.64	15	explicit linear model	multileveled self-organization
0.65	0.66	0.63	6	implicit nonlinear model	Neural Network (GA-MLP)
0.63	0.63	0.65	8	implicit nonlinear model	Neural Network (GA-MLP)
0.63	0.71	0.64	11	explicit nonlinear model	multileveled self-organization

$N = 275$ $N_{A,B} = 229$ $N_C = 46$ M: up to 1800

with

$R^2_{A,B,C}$ - R^2 calculated on the entire data set *D*

$Q^2_{A,B}$ - leave-one-out cross-validation on the data subset $D_{A,B}$

R^2_C - R^2 calculated on the test data subset D_C

m - number of variables used in the model

M - number of potential input variables; state space dimension

multileveled self-organization: High-dimensional modeling algorithm using multileveled self-organization with GMDH Networks as Active Neurons

Neural Network (GA-MLP): Genetic Algorithm for dimension reduction; Multilayer Percepton Neural Network for modeling

The *model type* column of table 1 distinguishes between implicit and explicit regression models. While Neural Networks typically distribute and hide the created model in the network the result of self-organizing GMDH Networks are explicit analytical models. Figure 11 shows, for example, the regression equation of the first model of table 1 and figure 12 lists a much more complex nonlinear model, exemplarily. Neither the formal model structure nor the input variables composition was given a priori; the model is completely self-organized. This true knowledge extraction from data has proven very useful and advantageous for model interpretation, evaluation, and implementation issues. So it is possible to implement these types of models in a MS Excel sheet, automatically, for immediate use for further analysis, evaluation, or just application purposes [16].

LC_{50} *(trout) [mmol/l]* = - 1.6023X1254 - 1.530X466 - 1.3148X1211 - 27.1340X481 - 0.8957X77 + 2.1469X994 - 0.2699X150 + 0.7736X1009 - 0.0313X355 + 5.8706X886 + 28.220

LC_{50} *(trout) [mmol/l]* = - 1.6023 $(\text{C-031})^{-1}$ - 1.53 MATS3e - 1.3148 $(\text{nOH})^{-1}$ - 27.1340 GATS3m - 0.8957 nxch3 + 2.1469 $(\text{SEigZ})^{-1}$ - 0.2699 LogDpH7 + 0.7736 $(\text{D/Dr09})^{-1}$ - 0.0313 D/Dr03 + 5.8706 $(\text{Mp})^{-1}$ + 28.220

Figure 11. Self-organized linear regression model in mathematical and chemical notation, correspondingly

```
---------------------------------------------------------------------
LC50 (trout) [mmol/l] = - 0.429829719X836 + 1.64397Z1
+ 0.9696540988X836X836 - 2.37319257
---------------------------------------------------------------------
---------------------------------------------------------------------
Z1= + 0.0051073216X994 + 0.064870976X4
+ 0.7824600925Z2 + 0.0377659874X101
- 0.392718518X101X101 + 0.0027544187X4X994
+ 0.0332232211X994Z2 + 0.001603542X101X994
+ 0.0011547157X4X4 + 0.0278558781X4Z2
+ 0.0013444835X4X101 + 0.1679958781Z2Z2
+ 0.0162168781X101Z2 - 0.01669142X101X101X994
- 0.013994861X4X101X101 - 0.168803X101X101Z2
- 0.008X101X101X101 + 0.042X101X101X101X101
- 0.387126711
---------------------------------------------------------------------
---------------------------------------------------------------------
Z2= - 0.15843533X188 + 0.6116860395Z3 + 0.500426Z4
- 0.277661247X188Z3 - 0.021393058
---------------------------------------------------------------------
---------------------------------------------------------------------
Z3= - 0.001649738X511 - 11.5528634X54
+ 0.0014663159X6X511 - 0.055364358X54X1129
+ 4.63388E-5X1129X1129 + 0.00603067X54X511
+ 5.2053X54X54 + 2.89005587E-5X54X511X1129
- 2.41891E-8X511X1129X1129 - 0.0053X6X54X511
+ 0.049891X54X1129 - 4.175796775E-5X54XX1129
- 2.5687E-5X6X54X511X1129
+ 2.149E-8X6X511X1129X1129
+ 1.1954591419E-4X54X54X1129X1129
- 2.00115163E-7X54X1129X1129X1129
+ 8.3746229977E-11X1129X1129X1129X1129
+ 6.4337698703
---------------------------------------------------------------------
---------------------------------------------------------------------
Z4= + 2.4932510741X958 + 1.16291Z5
- 2.286586535X958X958 - 0.313925295
---------------------------------------------------------------------
---------------------------------------------------------------------
Z5= - 0.135329249X6 - 0.231823342X932
+ 0.1968231776X988 - 0.135328623X6X62
+ 0.0068569682X932X932 - 0.011643437X932X988
+ 0.0049427686X988X988 + 1.4777933918
---------------------------------------------------------------------
```

Figure 12. Complex self-organized nonlinear regression model. Model structure, complexity, and variables composition is *not* defined a priori.

The results of the best individual QSAR models for the Daphnia, Oral Quail, Dietary Quail, and Bee data sets are summarized in tables 2 to 5.

Given the very high noise variance in the experimental toxicity data and the high chemical diversity of the compounds the data sets are composed of with respect to chemical classes and mode of action, the results of the individual models are indeed satisfactory, already, also compared to results of similar but more homogenous, i.e., simpler to model data sets.

Table 2. Five best models for the data set D_2 – Daphnia - with respect to $R^2_{A,B,C}$

$R^2_{A,B,C}$	$Q^2_{A,B}$	R^2_C	*m*	*model type*	*DM-method*
0.67	0.79	0.52	10	implicit nonlinear model	Neural Network (GA-MLP)
0.66	0.76	0.59	15	implicit nonlinear model	Neural Network (GA-MLP)
0.65	0.73	0.65	11	implicit nonlinear model	Neural Network (GA-MLP)
0.6	0.71	0.44	13	explicit linear model	multileveled self-organization
0.59	0.7	0.48	12	explicit linear model	multileveled self-organization

N = 258 $N_{A,B}$ = 215 N_C = 43 M: up to 2050

Table 3. Five best models for the data set D_3 – Oral Quail - with respect to $R^2_{A,B,C}$

$R^2_{A,B,C}$	$Q^2_{A,B}$	R^2_C	*m*	*model type*	*DM-method*
0.74	0.9	0.49	10	implicit nonlinear model	Neural Network (GA-MLP)
0.61	0.79	0.17	15	explicit linear model	multileveled self-organization
0.61	0.75	0.34	5	implicit nonlinear model	Neural Network (GA-MLP)
0.6	0.77	0.22	10	implicit nonlinear model	Neural Network (GA-MLP)
0.59	0.76	0.2	14	explicit linear model	multileveled self-organization
N = 104	$N_{A,B}$ *= 95*	N_C *= 19*	*M: up to 2100*		

Table 4. Five best models for the data set D_4 – Dietary Quail - with respect to $R^2_{A,B,C}$

$R^2_{A,B,C}$	$Q^2_{A,B}$	R^2_C	*m*	*model type*	*DM-method*
0.71	0.83	0.62	10	explicit nonlinear model	multileveled self-organization
0.69	0.87	0.27	21	explicit nonlinear model	multileveled self-organization
0.64	0.8	0.32	23	explicit linear model	multileveled self-organization
0.57	0.73	0.16	12	explicit linear model	multileveled self-organization
0.56	0.73	0.12	17	explicit linear model	multileveled self-organization
N = 118	$N_{A,B}$ *= 98*	N_C *= 20*	*M: up to 2100*		

Table 5. Five best models for the data set D_5 – Bee - with respect to $R^2_{A,B,C}$

$R^2_{A,B,C}$	$Q^2_{A,B}$	R^2_C	*m*	*model type*	*DM-method*
0.65	0.7	0.8	7	implicit nonlinear model	Neural Network (GA-MLP)
0.65	0.76	0.52	18	explicit linear model	multileveled self-organization
0.64	0.73	0.61	11	explicit linear model	multileveled self-organization
0.62	0.7	0.61	13	explicit linear model	multileveled self-organization
0.52	0.55	0.69	10	explicit linear model	PLS
N = 102	$N_{A,B}$ *= 85*	N_C *= 17*	*M: up to 2100*		

PLS: VIP parameter in PLS for dimension reduction; PLS for modeling (using the SIMCA implementation of PLS in both cases)

5.2. Combined models

All methods of automatic model selection lead to a single "best" model while the accuracy of model result depends on the variance of the data. A common way for variance reduction is aggregation of similar model results following the idea: Generate many versions of the same predictor/classifier and combine them in a second step. If modeling aims at prediction, it is helpful to use alternative models that estimate alternative forecasts. These forecasts can be combined using several methods to yield a composite forecast of a smaller error variance than any of the models have individually. The desire to get a composite forecast is motivated by the pragmatic reason of improving decision-making rather than by the scientific one of seeking better explanatory models. Composite forecasts can provide more informative inputs for a decision analysis, and therefore, they make sense within decision theory, although they are often unacceptable as scientific models in their own right, because they frequently represent an agglomeration of often conflict theories.

Based on the five sets of individual models that now serve as input information, we generated a combined model for each data set by a self-organizing GMDH Network algorithm. The result is five self-selected, optimally composed linear or nonlinear regression models, including their regression equation. It is shown from table 6 that the overall model performance for all 5 data sets increases sufficiently. Figures 13 to 17 plot the combined models for the five data sets correspondingly.

Table 6. Model performance summary of five combined models

data set	$R^2_{A,B,C}$	$Q^2_{A,B}$	R^2_C	*m*	*models*
D_1 - Trout	0.74	0.77	0.56	7	NN(1), F-NN(1), MSO(5)
D_2 - Daphnia	0.81	0.84	0.62	7	NN(2), F-NN(2), PLS(1), MSO(2)
D_3 - Oral Quail	0.84	0.9	0.53	4	NN(3), PLS(1)
D_4 - Dietary Quail	0.85	0.88	0.71	7	PLS(1), MSO(6)
D_5 - Bee	0.8	0.8	0.78	5	NN(2), MSO(3)

with
$R^2_{A.B.C}$ - R^2 calculated on the entire data set D_i
$Q^2_{A.B}$ - leave-one-out cross-validation on the data subset $D_{i,A.B}$
R^2_C - R^2 calculated on the test data subset $D_{i.C}$
m - number of models implemented in the combined model

models column:	The modeling method a model was generated with followed by the number of models of this type used in the combined model
	NN - Neural Network
	F-NN - Fuzzy Neural Network
	PLS - Partial Least Squares
	MSO - Multilayered Self-organization

It should be noted that the combined models are not just a composition, or the mean, of the five or seven best individual models of a data set but are an a priori unknown, optimal mix of models that – combined – decrease the error variance of the combined model most.

However, every individual or combined model is not able to also reflect the uncertainty given by the initial experimental toxicity data. Here the idea of a prediction interval seems useful.

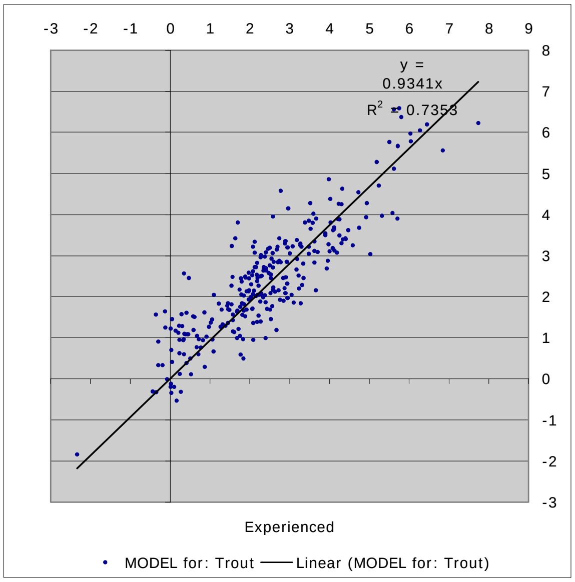


Figure 13. Scatter plot of the combined model for data set D_1 – Trout

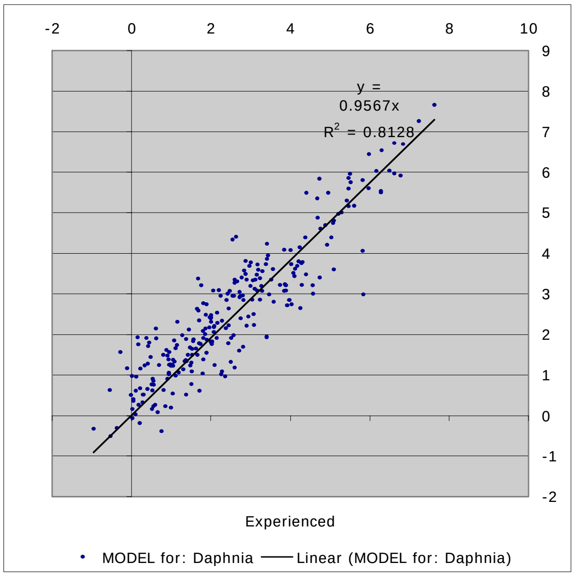


Figure 14. Scatter plot of the combined model for data set D_2 – Daphnia

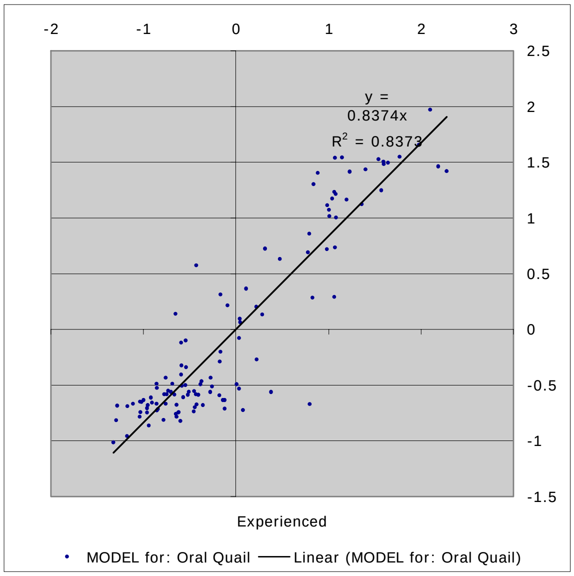


Figure 15. Scatter plot of the combined model for data set D_3 – Oral Quail

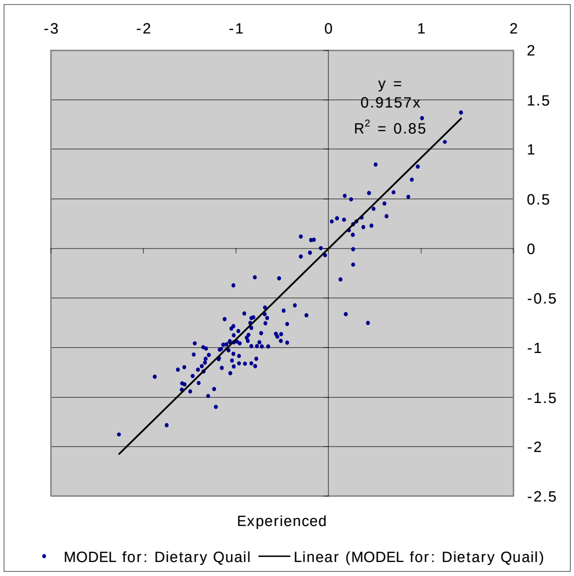


Figure 16. Scatter plot of the combined model for data set D_4 – Dietary Quail

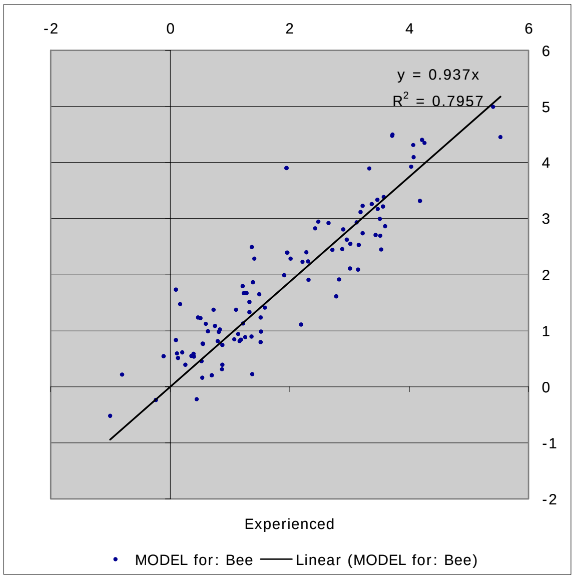


Figure 17. Scatter plot of the combined model for data set D_5 – Bee

5.3. Model uncertainty and prediction interval

As pointed out in section 2, toxicity data are highly noisy and therefore require adequate modeling, results interpretation, and decision support methods. Additionally, all methods of automatic model selection lead to a single "best" model. On this base are made conclusions and decisions as if the model was the true model. However, this ignores the major component of uncertainty, namely uncertainty about the model itself. In toxicity modeling it is not possible that a single crisp prediction value can cover and reflect the uncertainty given by the initial object's data. If models can be obtained in a comparing short time it is useful to create and apply several alternative reliable models on different data subsets or using different modeling methods and then to span a prediction interval from the models' various predictions for describing the object's uncertainty more appropriately. In this way a most likely, a most pessimistic (or most save prediction from a toxicity point of view), and a most optimistic (or least save) prediction is obtained, naturally, based on the already given models only, i.e., no additional (statistical) model has to be introduced for confidence interval estimation, for example, which would had to make some new assumptions about the predicted data, and therefore, would include the confidence about that assumptions, which, however, is not known a priori.

A prediction interval has two implications:

1. The decision maker is provided a set or range of predicted values that are possible and likely

representations of a virtual experimental animal test including the uncertainty once observed in corresponding past real-world experiments. The decision maker can base its decision on any value of this interval according to importance, reliability, safety, impact or effect or other properties of the actual decision to make. This keeps the principle of freedom of choice for the decision process.

2. Depending on which value is actually used, a prediction interval also results in different overall model quality values like R^2, starting from the highest accuracy for most likely predictions.

Figure 18 displays the prediction intervals for test set compounds (D_C) obtained from the predictions of the individual models contained in the combined model for the data set D_1 as reported in table 6.

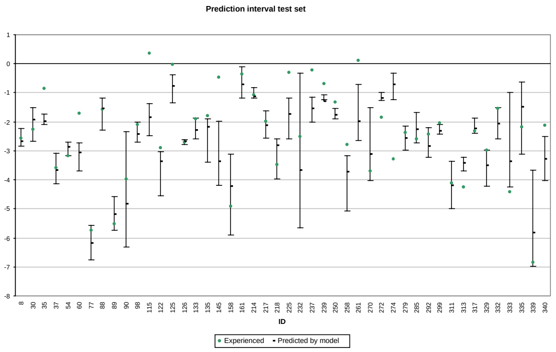


Figure 18. Prediction interval for test set compounds of data set $D_{1,C}$

In a real-world application scenario evaluation and decision-making can only base on predictions; no experienced toxicity value is given, usually, except those available from past tests. A supplement to providing prediction intervals that covers model uncertainty for decision making from another perspective can be the following approach:

1. For N compounds create a list of pairs $(y_i, \hat{y}_i)$ with y_i as the observed toxicity for a compound i and $\hat{y}_i$ as the predicted toxicity for the same compound i. N preferably equals the total number of compounds available for a data set, i.e., learning and testing data. The estimated/predicted values $\hat{y}_i$ can be any values of the prediction interval, minimum, maximum, mean, for example.
2. Sort matrix $\left[\mathbf{y}\ \hat{\mathbf{y}}\right]$ with respect to column $\hat{\mathbf{y}}$.
3. Create q equidistant classes based on $\hat{\mathbf{y}}$.

The result is q disjoint classes of corresponding observed and estimated toxicity values. For each class j, j=1, 2, .., q, the estimated toxicity mean and the minimum, maximum, and mean of the observed toxicities can be calculated. This means that here an interval of observed toxicity values for a given interval of predicted toxicities is obtained that describes the prediction's uncertainty for a related class or interval. Using a new compound's most likely prediction from the prediction interval, for example, this value would decide in which prediction class the compound would fit into along with the class' uncertainty given by the interval of past experienced toxicity values. Figure 19 plots the results of a derived decision model for q=12 classes from the predictions of the combined model of the trout data set D_1 reported in table 6 and table 7 lists the underlying data of figure 19, for reference.

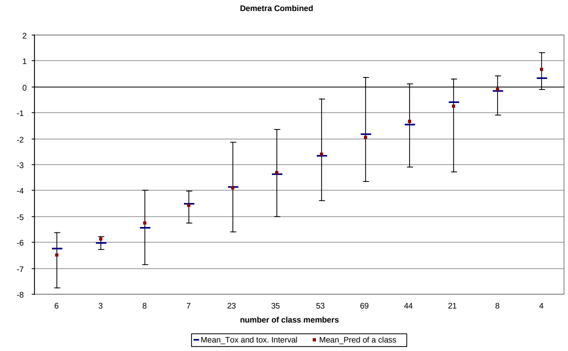


Figure 19. Decision model based on the predictions of the combined model of data set D_1 - Trout

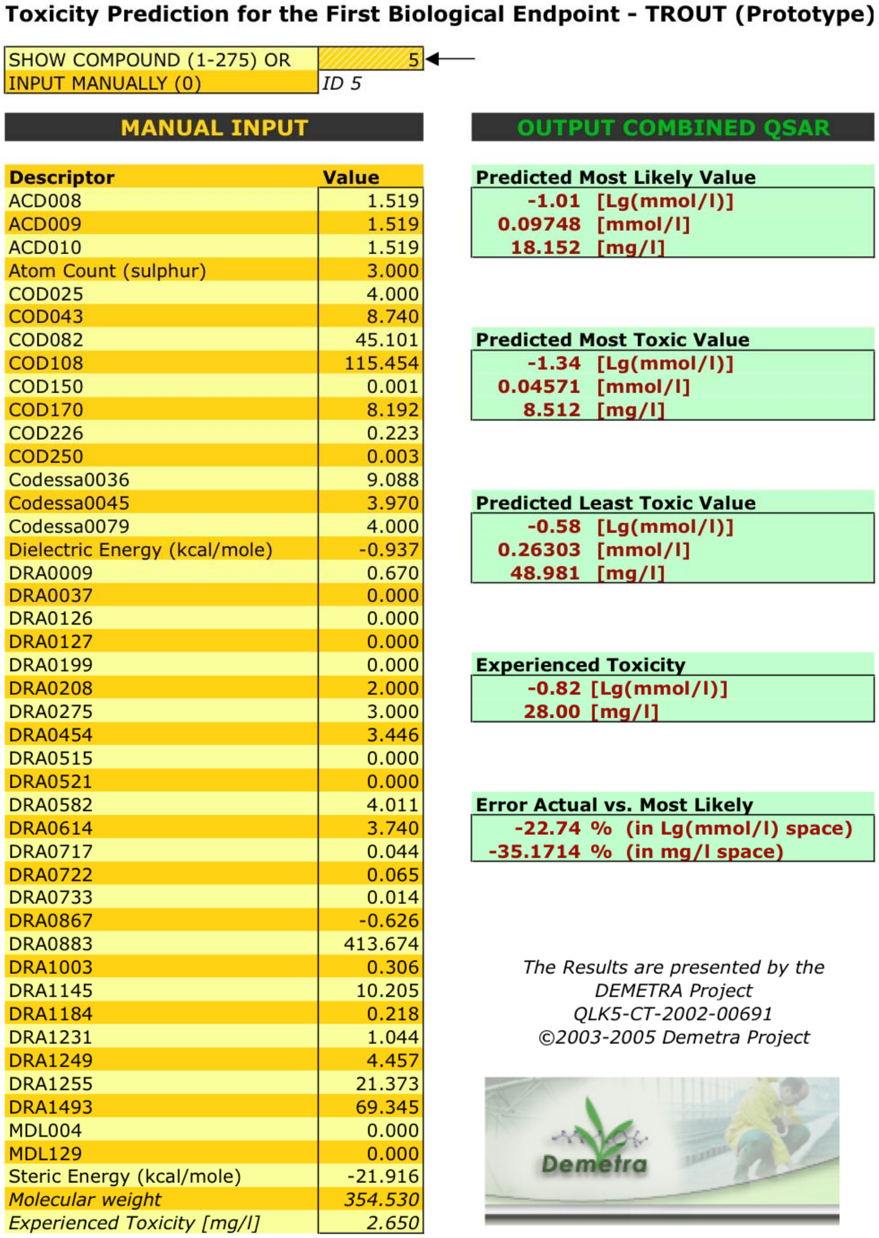
Table 7. Underlying data of the decision model displayed in figure 19

Class	*From*	*To*	*Min.*	*Mean*	*Max.*	*Mean*
	predicted toxicity		*observed toxicity*			*pred. toxicity*
1	-6.9	-6.24	-7.74	-6.23	-5.62	-6.49
2	-6.24	-5.59	-6.27	-6.03	-5.79	-5.88
3	-5.59	-4.93	-6.84	-5.45	-3.98	-5.26
4	-4.93	-4.27	-5.24	-4.5	-4.02	-4.57
5	-4.27	-3.61	-5.58	-3.87	-2.13	-3.89
6	-3.61	-2.95	-5.02	-3.37	-1.64	-3.31
7	-2.95	-2.29	-4.4	-2.66	-0.47	-2.61
8	-2.29	-1.63	-3.66	-1.83	0.36	-1.95
9	-1.63	-0.97	-3.1	-1.46	0.12	-1.33
10	-0.97	-0.31	-3.27	-0.6	0.3	-0.74
11	-0.31	0.35	-1.09	-0.15	0.43	-0.1
12	0.35	1.01	-0.1	0.32	1.33	0.66

6. Decision support model implementation

The data-driven concept for developing adequate toxicity prediction and decision models outlined in this paper to be used as alternative, substituting tools for animal tests during the projected extended evaluation of existing chemical compounds is implemented, exemplarily, for the trout data set in Microsoft Excel. This prototype is a fully working toxicity prediction tool that works on both any single compound of the given data set D_1 and any new compound when the required descriptor values for this compound are provided. The result is a most likely toxicity value in two common toxicity data spaces – mmol/l and mg/l – along with the prediction uncertainty expressed by the compound’s predicted highest and lowest toxicity, displayed numerically and graphically.

Figure 20 and figure 21 show the interface of this tool.



Toxicity Prediction for the First Biological Endpoint - TROUT (Prototype)

SHOW COMPOUND (1-275) OR	5
INPUT MANUALLY (0)	*ID 5*

MANUAL INPUT

Descriptor	Value
ACD008	1.519
ACD009	1.519
ACD010	1.519
Atom Count (sulphur)	3.000
COD025	4.000
COD043	8.740
COD082	45.101
COD108	115.454
COD150	0.001
COD170	8.192
COD226	0.223
COD250	0.003
Codessa0036	9.088
Codessa0045	3.970
Codessa0079	4.000
Dielectric Energy (kcal/mole)	-0.937
DRA0009	0.670
DRA0037	0.000
DRA0126	0.000
DRA0127	0.000
DRA0199	0.000
DRA0208	2.000
DRA0275	3.000
DRA0454	3.446
DRA0515	0.000
DRA0521	0.000
DRA0582	4.011
DRA0614	3.740
DRA0717	0.044
DRA0722	0.065
DRA0733	0.014
DRA0867	-0.626
DRA0883	413.674
DRA1003	0.306
DRA1145	10.205
DRA1184	0.218
DRA1231	1.044
DRA1249	4.457
DRA1255	21.373
DRA1493	69.345
MDL004	0.000
MDL129	0.000
Steric Energy (kcal/mole)	-21.916
Molecular weight	*354.530*
Experienced Toxicity [mg/l]	*2.650*

OUTPUT COMBINED QSAR

Predicted Most Likely Value
-1.01 [Lg(mmol/l)]
0.09748 [mmol/l]
18.152 [mg/l]

Predicted Most Toxic Value
-1.34 [Lg(mmol/l)]
0.04571 [mmol/l]
8.512 [mg/l]

Predicted Least Toxic Value
-0.58 [Lg(mmol/l)]
0.26303 [mmol/l]
48.981 [mg/l]

Experienced Toxicity
-0.82 [Lg(mmol/l)]
28.00 [mg/l]

Error Actual vs. Most Likely
-22.74 % (in Lg(mmol/l) space)
-35.1714 % (in mg/l space)

The Results are presented by the
DEMETRA Project
QLK5-CT-2002-00691
©2003-2005 Demetra Project

Figure 20. Interface of the implemented decision support model for predicting a chemical compound’s toxicity on the biological endpoint trout – page 1.

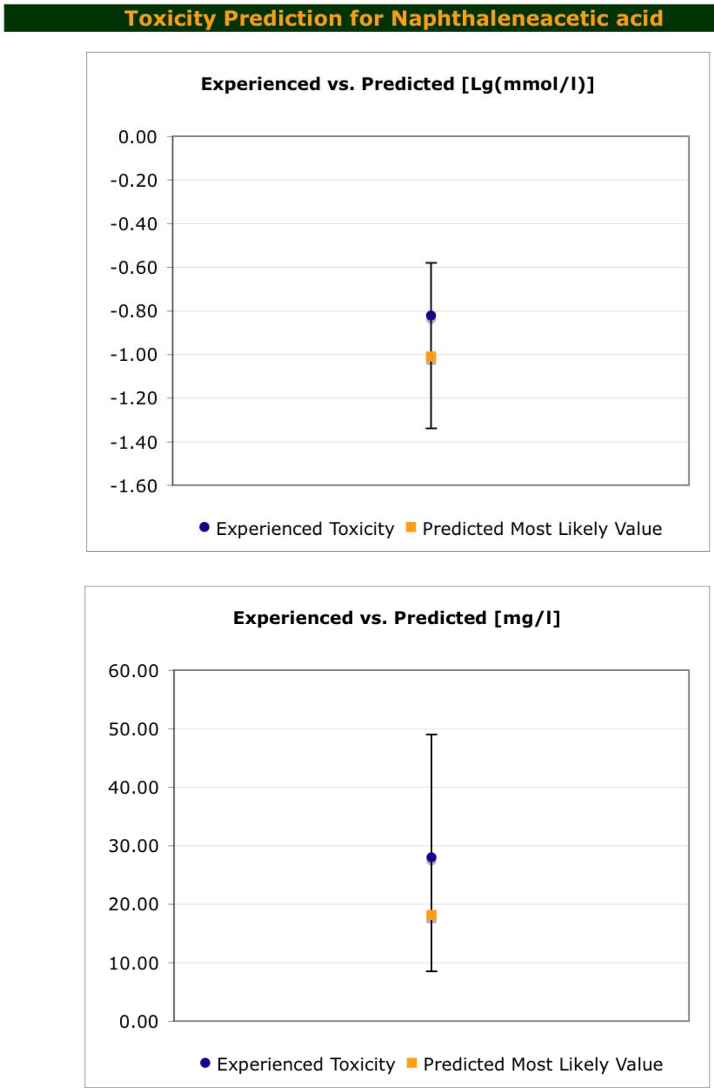

Toxicity Prediction for Naphthaleneacetic acid
Experienced vs. Predicted [Lg(mmol/l)]
0.00
-0.20
-0.40
-0.60
-0.80
-1.00
-1.20
-1.40
-1.60
● Experienced Toxicity
■ Predicted Most Likely Value
Experienced vs. Predicted [mg/l]
60.00
50.00
40.00
30.00
20.00
10.00
0.00
● Experienced Toxicity
■ Predicted Most Likely Value

Figure 21. Interface of the implemented decision support model for predicting a chemical compound's toxicity on the biological endpoint trout – page 2.

Some features, which are relevant for the specific purposes of this tool, should be noted here. Our approach was driven by the overall goal of providing a tool for regulatory use of QSAR models. A major problem with currently published QSAR models, from a regulatory point of view, is that they are much closer to a research tool than to a practical tool. In particular, they are sensitive to the human experience of the individual researcher. Typically, a researcher with a specific skill is using those more research-oriented models. And, it is expected that on the basis of her/his experience different results may be obtained. Such a situation, which is common in the research field, is neither a most favorable nor an acceptable case for regulatory uses during the authorization process of industrial chemicals, if different results can be expected depending on the person who is using the QSAR model. The tool we present here, vice versa, is tailored for regulatory uses, because it calculates a unique output value from the model, along with its uncertainty. The user does not require any particular experience in the QSAR model itself. However, she has to calculate the chemical descriptors indicated in the tool using certain publicly available software, but no further experience in QSAR modeling is needed.

7. Summary

In this paper we outlined a concept for developing alternative tools for toxicity prediction of chemical compounds to be used for evaluation and authorization purposes of public regulatory bodies to help minimizing animal tests, costs, and time associated with registration and risk assessment processes.

We started with a system theoretical analysis of ecotoxicological systems to clearly identify the problems to solve:

1. Animal tests as the source of toxicity data for QSAR modeling are described by a complex, nonlinear dynamic ecotoxicological system. The mortality rate of a certain species as the observed output variable of this system, however, is not object of toxicity QSAR modeling, directly. Instead, an input variable of the system – the external disturbance LC_{50} – is modeled by a pollutant's molecular structure. The system's observed output variable, the mortality rate y, is mapped by a single pair of observations (LC_{50}, y), which presumes a static linear relationship between these two variables a priori. The toxicity QSAR modeling problem, finally, transforms to building static, linear or nonlinear models. This, all together, is a strong simplification of the ecotoxicological system and adds uncertainty to results.
2. Toxicity data is very noisy due to a biological species' natural variability and due to the uncertainty of the animal test procedure. Also, there is not a single valid toxicity value but a certain range of experienced toxicities for a given chemical compound that can be seen all as true, reliable values.
3. Toxicity QSAR modeling is an ill-defined and high-dimensional modeling problem that requires adequate modeling tools.
4. Decision support has to take into account the uncertainty of the underlying system and the models.

Within the DEMETRA project, we generated five data sets for five biological endpoints that show very high quality. This quality feature refers to the reliability of the experimental toxicity data derived from past animal tests as well as to the calculation of molecular descriptors for the pesticides under study.

We addressed the problem of high-dimensional modeling of an ill-defined system by introducing multileveled self-organization, which incorporates state space dimension reduction, variables selection, data mining, and model evaluation into a single, autonomously running algorithm. We paid special attention to model validation and we suggested and implemented a two-stage model validation idea, which is composed of applying cross-validation and an algorithm's identified noise sensitivity, subsequently.

We combined several individual QSAR models to model ensembles that all show significantly increased model accuracy and, in addition, we assigned to every single prediction of a given compound a prediction interval to describe uncertainty.

Finally, this concept is implemented exemplarily in Microsoft Excel for real-world application.

A future work to do is the definition of standards for toxicity data, toxicity QSAR modeling, and model validation for improving reproducibility, transparency and acceptability of data-driven toxicity prediction tools to be established as a real alternative to animal tests.

[1] European Commission: White Paper. Strategy for a future Chemicals Policy, 27.02.2001

[2] European Commission: REACH in brief, 15.09.2004

[3] van der Jagt, K., Munn, S., Tørsløv, J., de Bruijn, J. (editors): Alternative approaches can reduce the use of test animals under REACH. Institute for Health and Consumer Protection, European Commission, *Joint Research Centre, Report EUR 21405 EN*, Ispra, 2004

[4] Müller, J.-A., Lemke, F.: *Self-Organising Data Mining. Extracting Knowledge From Data*, BoD, Hamburg, 2000

[5] Gini, G., Lorenzini, M., Benfenati, E.: Predictive Carcinogenicity: A model for Aromatic Compounds with Nitrogen-Containing Substituents Based on Molecular Descriptors Using Artificial Neural Network, *Journal of Chem. Inform. And Comp. Sci.*, (39)1999(6), pp. 1076-1080

[6] Müller, J.-A.: *Systems Engineering*. FORTIS Wien, 2000

[7] Lemke, F., Müller, J.-A.: Benfenati, E.: Modelling and Prediction of Toxicity of Environmental Pollutants. *LNAI 3303* (Eds. J. A. Lopez et al), Springer, Berlin, Heidelberg 2004, pp. 221-234

[8] Roncaglioni, A., Benfenati, E., Boriani, E., Clook, M.: A Protocol to Select High Quality Datasets of Ecotoxicity Values for Pesticides. *Journal of Environmental Science and Health*, Part B, B39, 641-652, 2004.

[9] DEMETRA, EC project, http://www.demetra-tox.net, 2004

[10] Farlow, S. J. (ed.): *Self-Organizing Methods in Modeling: GMDH-Type Algorithm*, Marcel Dekker, New York, 1984

[11] Barron, A. R., Barron, R. L.: Statistical learning networks: a unifying view, *Proceedings of the 20th Symposium Computer Science and Statistics*, 1988, pp. 192-203

[12] Ivakhnenko, A.G., Mueller, J.-A.: Parametric and nonparametric procedures in experimental systems analysis, *SAMS*, 9(1992), pp. 157-175

[13] Beer, S. T.: *Cybernetics and Management*, English University Press, London, 1959

[14] Lemke, F., Müller, J.-A.: Validation in self-organising data mining, *Proceedings ICIM 2002*, Lvov, http://www.knowledgeminer.com/pdf/validation.pdf

[15] Lemke, F.: Does my model reflect a causal relationship? *http://www.knowledgeminer.com/isvalid.htm,* 2002

[16] KnowledgeMiner: Self-organizing data mining and prediction tool, *http://www.knowledgeminer.com*

External Search Term Marketing Program
A Return on Investment Approach

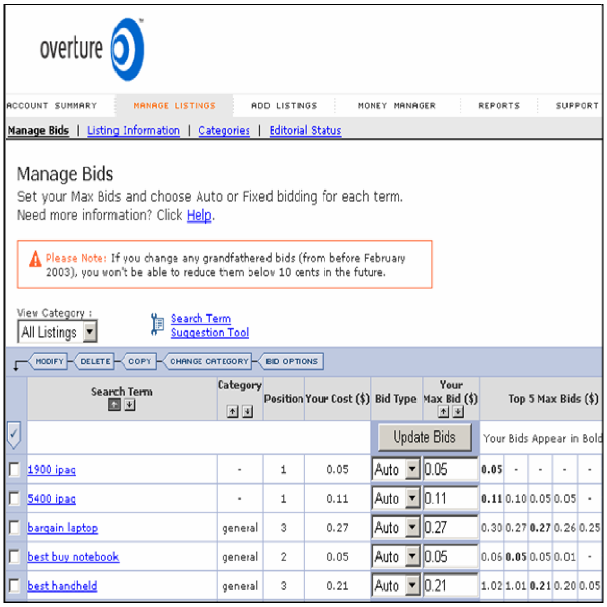
Pramod Singh, Laksminarayan Choudur, Alan Benson and Manoj Mathew
Hewlett-Packard Company
14231 Tandem Blvd, Austin, TX - 78727
pramod.singh[ατ]*hp.com*

Abstract

Buying prominent positions on Search engines' sponsored link space is a challenging job for an online marketer. It is not just about how many clicks these sponsored links generate, but the revenue and their life time value associated with them While calculating profitability of a few search terms may not sound that difficult, keeping track of profitability due to hundreds of search terms on a periodic basis is intractable, Once a system is built to compute 'Return on Investment' of search terms on a periodic basis, the next step is to analyze their performance. Improving landing pages, optimizing titling strategies and determining most cost effective positions for each term further improves the overall 'Return on Investment'. By implementing simple a decision tree modeling approach, we were able to reduce cost to revenue ratio from 108% to 47%, whilst increasing clicks 34%.

1. Introduction

When static online banner ads and internal promotional campaigns no longer remained competitive, online marketers came up with new strategies. One such strategy was to put sponsored links on search engines based on what visitors are searching for. These sponsored links appear on the top and the bottom of each search result page. (See fig 1.3). There could be more than one sponsored link for the term based on the term's popularity. Dell and Gateway are some of the major competitors of HP for bidding for the search term sponsored links on overture. Sponsored results links enabled Search engines to create a whole new space to showcase advertisers' offers relevant to visitors' interest and attracted them to go there before looking at actual web matches. Third party online advertising companies took advantage of this opportunity and created virtual market places to bid for search terms to get placed in higher position on sponsored links area of search engines. The most common and successful business model of these third party advertising companies was to charge for each click on sponsored link instead of charging for each sponsored link view.



Manage Bids

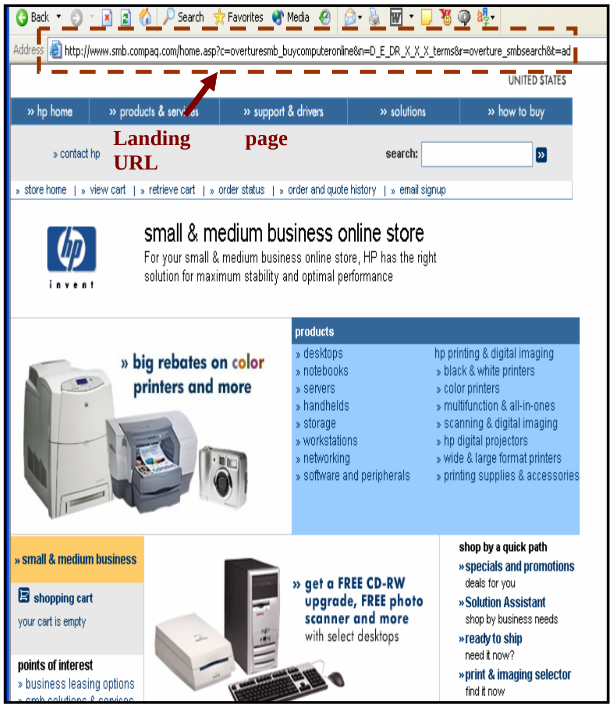
Set your Max Bids and choose Auto or Fixed bidding for each term.
Need more information? Click Help.

Please Note: If you change any grandfathered bids (from before February 2003), you won't be able to reduce them below 10 cents in the future.

	Search Term	Category	Position	Your Cost ($)	Bid Type	Your Max Bid ($)	Top 5 Max Bids ($)				
✓					Update Bids		Your Bids Appear in Bold				
□	1900 ipaq	-	1	0.05	Auto	0.05	**0.05**	-	-	-	-
□	5400 ipaq	-	1	0.11	Auto	0.11	**0.11**	0.10	0.05	0.05	-
□	bargain laptop	general	3	0.27	Auto	0.27	0.30	0.27	**0.27**	0.26	0.25
□	best buy notebook	general	2	0.05	Auto	0.05	0.06	**0.05**	0.05	0.01	-
□	best handheld	general	3	0.21	Auto	0.21	1.02	1.01	**0.21**	0.20	0.05

Figure 1.1: Overture online Bidding

Overture is one such third party vendor for managing sponsored links on various search engines like MSN, Yahoo and Lycos. Sponsored links are shown on top and bottom of search result pages of the search engine. Overture manages these sponsored links for advertisers interested in specific search terms. Advertisers pay to overture on advertisement performance basis. Thus, if in a week, a HP sponsored link for 'Desktops' on Yahoo was seen by 10,000 visitors and clicked by 200 visitors, overture charges HP for only 200 clicks. The cost per click to



Address http://www.smb.compaq.com/home.asp?c=overturesmb_buycomputeronline&n=D_E_DR_X_X_X_terms&r=overture_smbsearch&t=ad

small & medium business online store

For your small & medium business online store, HP has the right solution for maximum stability and optimal performance

Back
Search
Favorites
Media
UNITED STATES
» hp home
» products & services
» support & drivers
» solutions
» how to buy
» contact hp
Landing URL
page
search:
» store home | » view cart | » retrieve cart | » order status | » order and quote history | » email signup
invent
products
» desktops
» notebooks
» servers
» handhelds
» storage
» workstations
» networking
» software and peripherals
hp printing & digital imaging
» black & white printers
» color printers
» multifunction & all-in-ones
» scanning & digital imaging
» hp digital projectors
» wide & large format printers
» printing supplies & accessories
» big rebates on color printers and more
» small & medium business
shopping cart
your cart is empty
points of interest
» business leasing options
» get a FREE CD-RW upgrade, FREE photo scanner and more
with select desktops
shop by a quick path
» specials and promotions
deals for you
» Solution Assistant
shop by business needs
» ready to ship
need it now?
» print & imaging selector
find it now

Figure 1.2: Sponsored link landing page

be charged is decided by bidding among various advertisers for that search term. Bidding is done online on Overture website and bidders' can see their bidding price vis-à-vis other bidder's price (See Fig 1.1). The higher they bid, the higher position they get on sponsored link area. The starting bid for each search term is $0.10 and generally there is no restriction on number of sponsors or sponsored links for a search term. This project was done for search terms bought for HP SMB domain (smb.compaq.com) from overture in first half of 2003.

Each click on an HP online business store sponsored link goes to a landing page in HP online business store web site (See Fig 1.2). Each of these landing pages are tagged with Campaign parameters and they can be tracked in Ad-tracker tool (e-MOM tool) developed by HP e-Business Analytics team. This tool tracks number of total visits, unique visits, number of orders, order value, abandon rate and future orders for the traffic coming to these landing pages.

Objective

The primary objective for this project was to improve performance and maximize ROI of each search term bought by the HP online buiness store from Overture. In addition, we sought to maximize revenue subject to fixed spend per week for search term bidding.

Business Problem

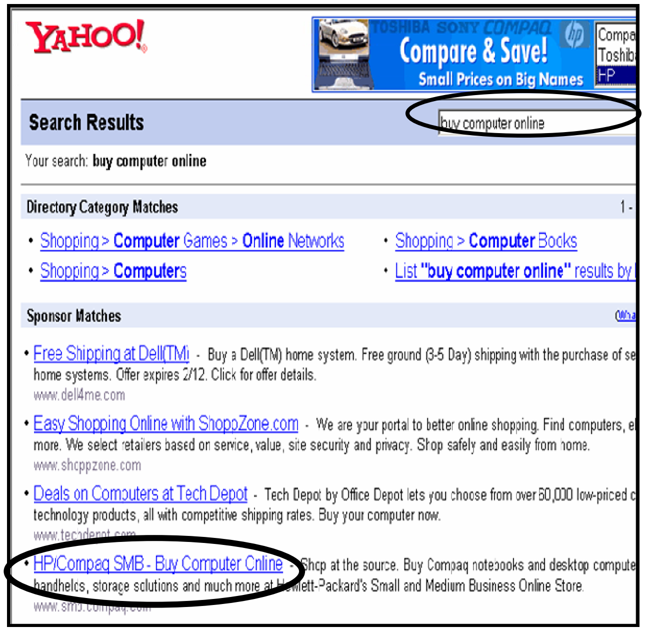
HP Business users responsible for buying and managing overture search terms had unique problems.

- How to track the performance of all search terms on a periodic basis?
- Which search terms to keep and which search terms to get rid of?
- How to maximize revenue generated from each term?
- Which new search terms to buy?
- How to maximize user experience of each sponsored link?
- How to decide the optimum position for each search term?

Observations

Following observations were made on the overture search terms at the start of the project:

1. Title and description were the same for each search term for the displayed results and it was not customized for individual search terms or group of search terms.
2. Some of the sponsored links were landing on pages which were not consistent with search terms. For example sponsored link for search term 'Cheap notebooks' was landing on a page, which had tablet pc pricing $1699 on top of it.
3. None of the landing pages were showing any promotions to users.
4. Some of the search terms were attracting lot of traffic but not generating any orders, resulting in heavy opportunity losses for HP.



Search Results

Your search: **buy computer online**

Directory Category Matches 1 -

- Shopping > **Computer** Games > **Online** Networks
- Shopping > **Computers**
- Shopping > **Computer** Books
- List "**buy computer online**" results by

Sponsor Matches

- Free Shipping at Dell(TM) - Buy a Dell(TM) home system. Free ground (3-5 Day) shipping with the purchase of se home systems. Offer expires 2/12. Click for offer details. www.dell4me.com
- Easy Shopping Online with ShoppZone.com - We are your portal to better online shopping. Find computers, el more. We select retailers based on service, value, site security and privacy. Shop safely and easily from home. www.shoppzone.com
- Deals on Computers at Tech Depot - Tech Depot by Office Depot lets you choose from over 60,000 low-priced c technology products, all with competitive shipping rates. Buy your computer now. www.techdepot.com
- HP/Compaq SMB - Buy Computer Online - Shop at the source. Buy Compaq notebooks and desktop compute handhelds, storage solutions and much more at Hewlett-Packard's Small and Medium Business Online Store. www.smb.compaq.com

Figure 1.2: Sponsored links

Approach

The team started the project with tackling the first issue that business users had: tracking performance of each search term on a weekly basis. The three basic and important metric of search term performance were a) Number of clicks, b) Number of Orders, c) Cost to revenue ratio. The source of data for these metrics are; 1) Overture online search term summary reporting 2) e-MOM tool developed by Analytics team which tracks performance of campaigns and Ads.

After identifying these data sources, the next step was to integrate them making sure that the data is defined in same way in both the sources. Landing pages for each of these sponsored links were coded to capture the campaign name and sub name indentified in the sponsored link URL. Unfortunately, the campaign names are not coded consistently in both the Overture search term bidding system, and HP's internal campaign tracking system (eMom) Because of this inconsistency there is a large amount of manual work involved in integrating the data sources. The team proposed some changes to url pattern of landing pages to reduce these inconsistencies,and thus eliminate the need for manual integration.

Following the integration, the list of below-mentioned metrics are computed:

i) Avg. cost -- Average cost of a click.

ii) Total Clicks (Overture) – Total clicks for search terms reported in Overture.

iii) Cost per Click – Cost per click for the search term.

iv) Avg. Position – Average position of search term during the time period.

v) Total Cost – Total cost paid for the search term.

vi) Total e-MOM visits – Total visits to the landing page as reported in e-MOM.

vii) Unique e-MOM visits – Unique visits to the landing pages as reported in e-MOM.

viii) Number of Orders – Total numbers of orders placed in landing page sessions as reported in e-MOM

ix) Order value – Total revenue generated from orders placed in landing page sessions as reported in e-MOM.

x) Abandon rate – Abandon rate from the landing pages of search term as reported in e-MOM.

xi) Future Orders (30 days,60 days,90 days,120 days) – Orders after Initial landing page session as reported in e-MOM.

Once the above metrics were generated, the next step was to develop a tool which can evaluate the performance of these search term and that can be used week after week without any effort.

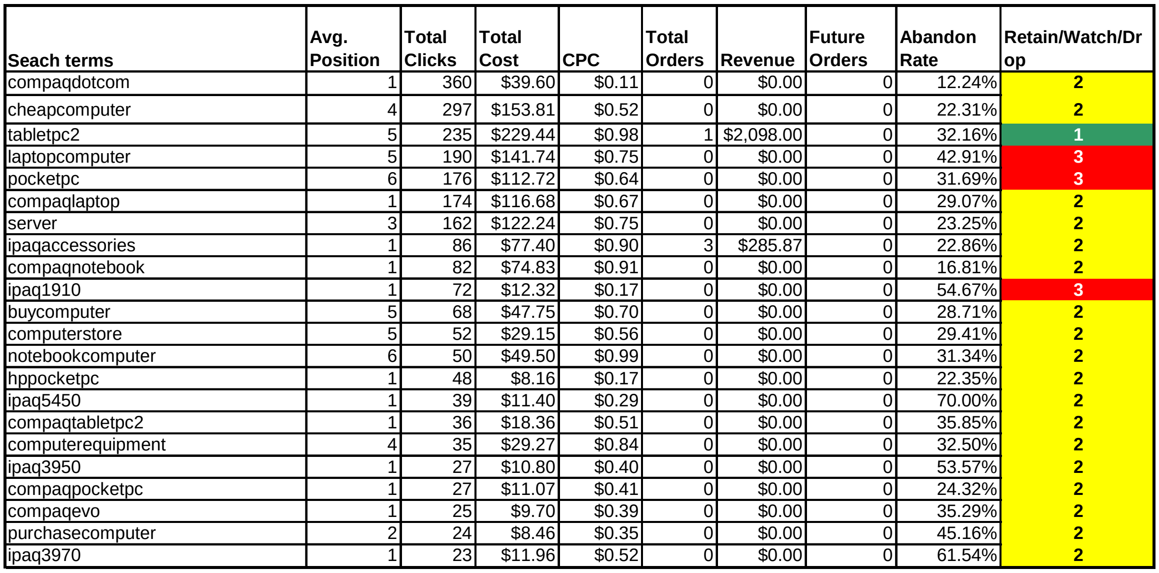
A Microsoft Excel based tool was developed to track the performance of each and every search term with a flexibility to define success criteria.

In addition to developing this tool, the team did research on search term sponsored links. The team looked at sponsored links for each sponsored links and recommended the ways to improve performance of those terms. Team also suggested some of the new terms that can be bought on Overture as explained later in the paper.

Tools developed

The team came up with Microsoft Excel based DSS (Decision Support System) or Decision Tree, as called by its users, which divides the list of all search term in to three categories: 1) Retain 2) Watch and 3) Drop. As and when the data for each search term is changed/refreshed, the macro associated with DSS runs and assigns each search term in to one of three above-mentioned buckets. Each group is represented with different color which makes it easier for users to identify the bucket (See example in Fig 1.4).

Results of DSS are based on the value of parameters defined in Success Definition Criteria (SCD) tool. The SCD tool works on decision tree logic. Each node of the tree is represented as decision criteria.



Seach terms	Avg. Position	Total Clicks	Total Cost	CPC	Total Orders	Revenue	Future Orders	Abandon Rate	Retain/Watch/Drop
compaqdotcom	1	360	$39.60	$0.11	0	$0.00	0	12.24%	2
cheapcomputer	4	297	$153.81	$0.52	0	$0.00	0	22.31%	2
tabletpc2	5	235	$229.44	$0.98	1	$2,098.00	0	32.16%	1
laptopcomputer	5	190	$141.74	$0.75	0	$0.00	0	42.91%	3
pocketpc	6	176	$112.72	$0.64	0	$0.00	0	31.69%	3
compaqlaptop	1	174	$116.68	$0.67	0	$0.00	0	29.07%	2
server	3	162	$122.24	$0.75	0	$0.00	0	23.25%	2
ipaqaccessories	1	86	$77.40	$0.90	3	$285.87	0	22.86%	2
compaqnotebook	1	82	$74.83	$0.91	0	$0.00	0	16.81%	2
ipaq1910	1	72	$12.32	$0.17	0	$0.00	0	54.67%	3
buycomputer	5	68	$47.75	$0.70	0	$0.00	0	28.71%	2
computerstore	5	52	$29.15	$0.56	0	$0.00	0	29.41%	2
notebookcomputer	6	50	$49.50	$0.99	0	$0.00	0	31.34%	2
hppocketpc	1	48	$8.16	$0.17	0	$0.00	0	22.35%	2
ipaq5450	1	39	$11.40	$0.29	0	$0.00	0	70.00%	2
compaqtabletpc2	1	36	$18.36	$0.51	0	$0.00	0	35.85%	2
computerequipment	4	35	$29.27	$0.84	0	$0.00	0	32.50%	2
ipaq3950	1	27	$10.80	$0.40	0	$0.00	0	53.57%	2
compaqpocketpc	1	27	$11.07	$0.41	0	$0.00	0	24.32%	2
compaqevo	1	25	$9.70	$0.39	0	$0.00	0	35.29%	2
purchasecomputer	2	24	$8.46	$0.35	0	$0.00	0	45.16%	2
ipaq3970	1	23	$11.96	$0.52	0	$0.00	0	61.54%	2

Figure 1.4: Results of DSS based on Success Criteria Definition (1 = Retain, 2 = Watch, 3 = Drop)

We start from the top of the tree and go down following the logic. The result of decision criterion at each node determines which way to follow down the tree. The bottom of the tree constitutes multiple roots. Each root has a specific value, which represent one of the three result buckets, Retain, Watch and Drop.

The DSS is designed to give total flexibility to user for binning the search terms. Users can easily change the criterion for each bin by changing the value of parameters in the SCD tool (See fig 1.5) and the DSS will automatically assign search terms in appropriate buckets according to new criterion defined. Flexibility provided in SCD tool to set thresholds for different attributes made sure search term chosen were significant and based on the performance defined by the user.

Analysis and Implementation

The data analysis had two separate aspects. First the DSS tool offered specific recommendations for bidding per search term. Second, a more thorough evaluation of the Overture search term cost data and HP's internal traffic and purchase data provided insight into why performance differed, and how ROI could be improved. The following points summarize broad areas of analyses and implementation:

Analysis of the cost and revenue data showed that bidding on some of the loss making terms was brought down to minimum level since the DSS isolated search terms with poor ROI and recommended dropping them. For example bidding on ‘Cheap Notebook’ and ‘Cheap Laptop’ were brought down from $0.70 and $0.65 per click to minimum possible level of $0.10 Total cost of these terms came down by more than 70% as a result of changes made to average position.

Deeper analysis suggested that poorly performing search terms could be improved by customizing titles and descriptions based on the search term. For example if users are searching for ‘Cheap computer’, the title and description should suggest the lowest prices of computers are available on the HP SMB store. Additionally, search terms emphasizing low prices should reference active promotions in the search results titles and descriptions to attract more potential buyers. Titles and descriptions of 12 search

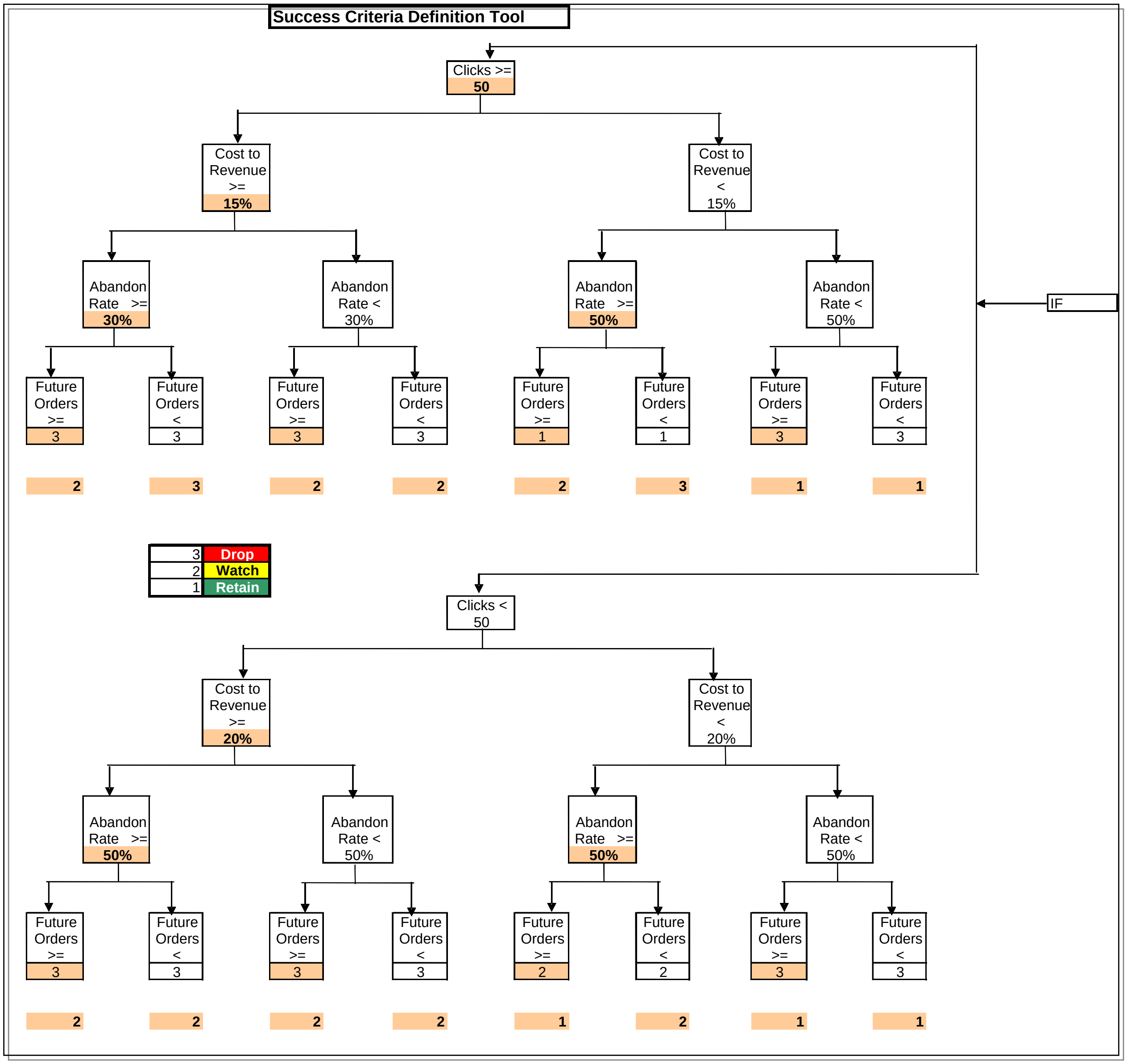


Figure 1.5: Success Criterion Definition tool.

terms were changed as a pilot test to determine whether or not they bring any change to performance of overture search terms.

An additional area for ROI improvement was the sponsored linked destinations. The sponsored links destination links should be linked to specific pages relating to the search term in question. For example terms consisting 'computer' or 'pc' in it should be taken to page, which shows desktops and notebooks (may be pocket pc too) instead of just taking them of HP online business store home page. The Cost to revenue ratio of these terms were 141% compared to 11% of other terms landing on the home page and less than 30% for overall search terms. These terms were costing as much as 20% of the total cost to the project with less than 2% share in revenues.

New search terms were suggested to business users that can be bid on overture.

These terms were suggested based on what users were searching for internally while in the HP online business store. Some of these terms were product specific and were not bid on overture site. Apart from this, the team also suggested few search term which were very common and had good potential to attract buyers. Some of the new terms proposed were 'memory', 'keyboard', 'business computers', and 'notebook accessories' etc.

Recommendation

The project's initial business objective was to increase revenue from external search term, but the team recommended rearticulating the objective to focus on costs as well. Accepting our recommendation, the objective was changed to "Optimize Return on Investment (ROI) from the external search terms". The Objective was changed to make sure that project does not generate losses while attempting to increase revenue. The team started reporting 'Cost to Revenue' number on a weekly basis as a measure of ROI. This brought lot of attention on increasing revenue while keeping costs low.

Use the DSS to determine which search terms to retain, drop or watch closely. External search term data should be collected and fed to the DSS tool on a monthly basis and decision should be used based on the results from the tool. The decision criteria in DSS tool can be changed by business users per their requirements.

The titles and descriptions of the search term in overture may be modified and improved to attract potential buyers. Existing promotions can be used to promote search terms. Also titles and descriptions should be modified and customized by the search term.

Landing page of the search terms should fit with the search term meaning. For example if a Search term with 'Cheap' word in it, is bought, we should ensure that landing page takes user to the cheap/cheapest products of the product category, he is looking for.

Use current and active promotions as part of Search result page's title and description. This would help attract customers who are more sensitive to price and promotions.

Promotions should also be shown to the visitors when they come to landing page of sponsored links. It may increase the probability of their buying on the HP online business store.

A new landing page should be designed for search terms having 'PC' and 'Computer' in it (but not having laptop, notebook, desktop or pocket pc) showing only these product categories instead of taking visitors to SMB home page. If for some reason, such page can not be designed, the bidding on such term should be decreased to lowest to avoid any further erosions of project bottom line.

Changes made and results observed

Based on the recommendation from the team, business owners made the following changes to the overture search terms.

1) Reduce bidding of few terms

Bidding price of 'Cheap Notebook' was reduced to bring the average position from 3rd position down to 12th position, bid on 'Cheap Laptop' was reduced to bring the average position from 4th position down to 9th position and bid on 'Cheap notebook computer' was reduced to bring the average position from 4th position down to 8th position. The immediate result of this change was that cost for these terms dropped down significantly from $120 a week to $16 a week (86 % drop) whereas there was no change in revenue or number of orders .

2) Change the title and description of few search terms.

The titles and descriptions of 12 search terms were changed to customize that according to the meaning of search terms. The following 12 search terms were changed:

1. Tablet PC
2. Compaq notebook
3. Compaq laptop
4. Server
5. Ipaq Accessories
6. Hp Pocket PC
7. Buy Computer
8. Cheap Computer
9. Compaq evo
10. Notebook computer
11. Laptop Computer
12. HP Pocket PC

For example, the following changes were made in the title and description of sponsored link for 'Tablet PC':

BEFORE:

HP/Compaq SMB - Tablet PC- Shop at the source. Buy Compaq notebooks and desktop computers, HP servers, handhelds, storage solutions and much more at Hewlett-Packard's Small and Medium Business Online Store.

.

AFTER:

Buy a Compaq Tablet PC - Experience a new kind of computing. This new and innovative pc is available at the Hewlett-Packard SMB store starting at $1,699.

We can see from the above example that titles and descriptions of 'Tablet PC' was very generic before the change. It was not indicating anything specific about the Tablet PC. But after the change, it was customized to make it search term specific, 'Tablet PC' in this case.

We observed a significant improvement in traffic coming to these sponsored links after the change. Total number of clicks went up from 1,995 to 2,680+ a week. Cost to revenue came down from 107% to 47% and Abandon rate came down from 34% to 27%. Revenue went up while cost remained constant during the period.

Campaign Metric	**Before**	**After**
Cost / Revenue	107%	47%
Clicks to Campaign	2,680	1,995
Abandonment Rate	34%	27%

Going Forward

The team suggests that overture search terms and HP's ad tracking tool should be made more synchronous so that data integration between two sources can be made smoother. This is very important since this will allow more frequent performance reporting of overture search terms. It will also avoid extra manual effort when number of search term on overture site is increased from less than two hundred to five- six hundred.

Findings of the current project can be used by other HP business units, which are bidding for search term sponsored links on overture. The learning of this project can also be used by business users for other third party advertising companies like Quigo, LookSmart, FindSmart, etc.

Special landing pages should be created (if not already present) for external search terms sponsored links, highlighting promotions on those pages.

Title strategy study should be expanded to other product lines such as desktops, notebooks and servers.

For terms having 'pc' and 'computer' landing page should be designed to have all types of 'pc' or 'computer' instead of taking users to home page. For example 'buy computer' search term should land to a page which has promotional offer both for desktop computer and for notebook computer. Currently these terms are taking users to SMB home page.

Weekly/monthly report containing terms yielding profits and those causing losses should be generated along with the cost incurred on those. This will help business owners keep check on expenses while promoting high revenue terms.

Optimal Allocation of Online Advertisements Using Decision Trees and Non-Linear Programming

David Montgomery
Poindexter Systems, Inc.
dmontgomery[$\alpha\tau$]poindextersystems.com

Abstract

Advertisers that promote their products online are turning towards more sophisticated statistical techniques for finding the right audience for their products while at the same time meeting business rule requirements. In an automated environment, the data necessary to accurately identify users and to find separation of preferences of one product over another can be challenging. An advertiser that has many products to promote will typically find that even after segmentation of their audience using regression based techniques such as decision trees will find that across all leaf nodes one product will have a higher response rate than all other products. The consequence for an advertiser is that it will be optimal to show one product all of the time to maximize response rates. However, from an advertiser perspective this solution can be suboptimal due to the fact that the remaining products must be advertised at a minimal level. The objective of this study is to demonstrate how non-linear programming techniques can be applied to decision trees to optimally override the default recommendations in order to optimally meet business objectives.

1. Introduction

With the increasing accountability that online marketing mangers face for generating measurable lift with online advertisements, more sophisticated analytic techniques for reaching there target audience are being employed. Marketers faced with promoting many products given limited resources typically employ segmentation techniques such as decision trees to identify the right product to advertise to the right potential customer in order to maximize a response measure such as clicks, registrations, revenue, or profit.

Provided that the probability estimates of the products in the leaf nodes of a decision tree for each product are stable and robust over time, the use of decision trees in theory will generate lift by optimally distributing product mixes to the appropriate audience. However, the decision tree recommended distribution of product mixes can lead to several problems for advertisers. 1) Only one product is recommended to all segments of a decision tree. For example given a decision tree with four products and twenty leaf nodes, it can be optimal to recommend product 1 to all audiences. 2) The decision tree recommended product mix does not meet the required product mix. For example, the default recommended product mix in order to maximize a response measure can lead to product 1 receiving 20% of all impressions while product 2 receives 80% of all impressions. However, the required marketing mix for product 1 and product 2 is for Product 1 to receive 40% of the impressions while product 2 receives 60% of the impressions.

The objective of the paper is to demonstrate how the default recommendations of a decision tree can be optimally overridden to allocate online advertising impressions in a customer anonymous environment order to meet the product mix requirements of online advertisers.

The use of constrained optimization for allocating online advertisements has been previously documented For further reading refer to [1], [2] and [3]. However, these approaches typically segment by slot. The unique contribution of this paper is a fully specified profit maximizing formula applied to the leaf nodes of multiple decision trees created per slot.

This paper is organized as follows. In section 2, a background on online advertisement data is discussed.

In section 3, issues regarding the creation of decision trees for online advertising data are discussed. Section 4 discusses the general constraint based profit maximizing formulation of a decision tree while section 5 discusses and application of constraint based optimization to decision trees.

2. Online Advertisement Data

A typical online advertising campaign for a Poindexter Client can typically server anywhere from 100,000 online impressions per week up to 1billion or more per week. At Poindexter Systems, we store in our database the historical record of each impression served for each product on each slot along with the data attributes of each impression.

In the anonymous web environment data attributes that are know with relative certainty are the slot id's of the impressions, i.e. on which web page and what location on the web page an advertisement impression is served. Also known is time of day, day of week along with the browser type, bandwidth and IP address. Many other data attributes can be further derived from IP address such as ZIP code which can then be further rolled up into more discrete geographic regions. ZIP code can further be mapped to PRIZM clusters and other geographic based data such as Census data.

Costs associated with the delivery of online advertisements can vary. Typical cost metrics in the ad serving industry are cost per clicks (CPC), cost per acquisition (CPA), and costs per 1000 impressions served (CPM). The cost measure used in this paper for deriving a profit maximizing formula will be CPM's. For a further discussion on advertising pricing mechanisms refer to [4].

3. Decision Tree Construction

The use of decision trees in online advertisement compared to other regression based techniques such as polychotoumous logistic regression and neural networks have many advantages such as automated variable selection, ease of interpretation, visualization of results, and the handling of categorical data without the need for binning. However, decision trees are known to have disadvantages such as unstable probability estimates [2]. Techniques have been developed to deal with the instability of probability estimates of decision trees such as voting based procedures most notably bagging and boosting, and smoothing techniques such as Laplace correction, and M-estimation smoothing. Because a single representation of a decision tree is required, probability estimates will be handled using M-estimation smoothing. For further reading on cost sensitive learning of decision trees and probability smoothing refer to [2] and [3].

Typically in online advertisement the click and registration rates are classified as rare events and follow a Poisson distribution. The latest developments in decision tree theory and cost sensitive learning suggest that under-sampling of rare events is not necessary and that decision tree splitting criteria such as entropy and gini are relatively robust against rare events [3] and [5]. Given these insights decision trees are grown using all of the data thus preserving the true probability estimates of a leaf node. Furthermore a separate decision tree is created for each slot.

4. General Constraint Based Profit Maximizing Formulation of a Decision Tree

The end result of decision tree construction will yield the following information. A separate decision tree for each slot, the number of leaf nodes identified for each tree, probability estimates for each category of the dependant variable within each leaf node, and the percentage size of the leaf node relative the size of the root node of each tree. Given this information along with price value for the dependant variable and costs as measured by CPM's per slot, a general constrained profit maximizing formula can be derived.

Notation used for the derivation of applying constrained optimization will be described as follows. Let s_i represent an individual slot within the total slots available S and let cpm_i represent the *CPM* for each slot s_i. Furthermore let I_i represent the total number of impressions forecasted for each s_i. For product specifications let p_j represent product j given P products and let v_j represent the price value for p_j. For leaf node specifications let n_{ji} represent an individual leaf node in N nodes for the decision tree created for s_i and let c_{ki} represent the size of the leaf node for node s_i. Let r_{ijk} represent the response rate for each product in each leaf node for each site. Finally, let w_{ijk} represent the product display weight for each product in each leaf node for each site. Because the product display weight represents the percentage of time that each product should be shown this implies that following bounds.

$$\text{For each } i, j, k \ 0 \leq w_{ijk} \leq 1 \quad (1)$$

The above bounded conditions imply that no product can be shown more then 100% of the time in a given leaf node and an explicit non-negativity constraint for each product display weight.

The product display weight represents the percentage of time that each product should be shown. It is the product display weight that represents the unknowns in the constrained optimization model. Given the above notations the general profit maximizing formula can now be derived.

Let revenue can be defined as follows:

$$\text{Max Revenue} = \sum_{i=1}^{S}\sum_{j=1}^{N}\sum_{k=1}^{P} I_i v_k r_{ijk} c_{ij} w_{ijk} \quad (2)$$

Let cost be defined as follows:

$$\text{Min Cost} = \sum_{i=1}^{S}\sum_{j=1}^{N}\sum_{k=1}^{P} \frac{I_i}{1000} cpm_{ik} c_{ij} w_{ijk} \quad (3)$$

From the definition of profit maximization let profit (π) be defined as

$$\text{Max } \pi = \text{Max Revenue} - \text{Min Cost} \quad (4)$$

Subject to:

For each i, j, k $0 \le w_{ijk} \le 1$ (5)

For each i, j $\sum_{k=1}^{P} w_{ijk} \le 1$ (6)

The linear constraint in (6) combined with the product display weight bounds in (5) forces the sum of the product display weight to be less than or equal to 1 and greater than or equal to 0. That is, no node can recommend products more than 100% of the time. Given these conditions, taking the default recommendation of a decision tree implies that a decision tree can also be interpreted as a solution to a bounded linear programming problem.

Once a decision tree can be represented as a constrained optimization problem, the default recommendations of a decision tree can then be optimally constrained to meet business objectives. The following will derive the constraints necessary to optimally reallocate the impressions served for each product. Let I_k represent the number of impressions that are allocated to be served to each product. I_k can be derived from the decision tree for each slot as follows.

$$I_k = \sum_{i=1}^{S}\sum_{j=1}^{N} I_i c_{ij} w_{ijk} \quad (6)$$

The impression ratio (IR) of each product can the be calculated as

$$IR_k = \frac{I_k}{\sum_{i=1}^{S} I_i} \quad (7)$$

Not derived in this paper but following the same lines of logic as deriving the impression ration constraint, many other constraints to optimally override the default recommendation of a decision tree can be derived. For example, product ratio constraints defined as of all the registrations each offer must meet a target ratio, registration goal constraints which mean that a minimum number of registrations must occur.

5. Application

In this section, optimally constraining a decision tree to override the default recommendations using non-linear programming is implemented. The data is from a live campaign but the company name and the product names have been changed for confidentiality reasons. The optimization is implemented using Poindexter Systems flag ship product that automates the construction of decision trees and non-linear programming. Specifications of the decision tree and optimization process are input into an ASP application with the front end built using ASP.NET and the back end completely powered by SAS base 9.2, SAS Stat, SAS EM 5.1, SAS Graph, and SAS OR. SAS processing is carried out on a multithreaded dual process Intel Zeon chip in a windows 2003 server environment.

In this case study, the client has twelve products that it would like to server on one slot. The objective of the campaign is to maximize registration rates subject to the constraint that each product must meet a minimum percentage of being shown. Figure 1 shows the minimum required threshold for each product.

The data collection period lasted for one week. Each impression that was served for each product along with each product registration was collected over the one week period along with attributes for each impression such as time of day, day of week, bandwidth, recentcy, frequency, and IP address derivable information such as geography and PRIZM codes.

A decision trees was created for the slot with the dependant variable being the registrations for all products listed in Figure 1 along with the non response category coded as NR. Figure 2 below shows for the one week data collection period the frequency break down of the number of impressions served and product registrations across the slot while Figure 3 provides a breakdown of the leaf node sizes by training and validation data sets.

Figure 4 shows the results of the unconstrained decision tree for each slot in terms of the number of leaf nodes for each decision tree, the forecasted impressions for each slot, and the expected impressions allocated to each product using the formula from (6) and (7).

Because the default recommendation of the decision tree does not meet the required impression allocations for each product, using the techniques described in section 4, non-linear programming was applied to find the product display weights for each site, each node, and each product to generate the required impression rations while still maximizing overall registrations. Registration maximization formula was derived from the general profit maximizing formula by setting the CPM's for each site to 0 and the value for all products to 1.

Once the solution was calculated, the model was placed live on our ad server. The criteria for success is for the decision tree to serve the products in the desired proportions. Figure 5 displays the predicted impression ratio served for each product versus actual impression delivery for each product.

6. Conclusion

It has been demonstrated that combining multiple decision trees in a constrained profit maximizing formulation can allow advertisers to optimally override the default recommendation of decision trees in order to meet business objectives. The optimal allocation of advertising impressions using the methodologies employed in this paper are dependant upon the degree of accuracy of the probability estimates comprising decision trees such as the probability estimates of the leaf nodes of a decision tree and the percentage size of the leaf nodes compared to the root node.

All techniques described in this paper are fully implemented in Poindexter Systems flag ship product POE in an ASP environment and allows for the automation of decision tree construction and constrained optimization in real time.

7. References

[1] A. Amiri, S. Menon, "Efficient Scheduling of Internet Banner Advertisements", ACM Transactions on Internet Technology, November 2003, pp. 334-346.

[2] D.M. Chickering, D. Heckerman, "Targeted Advertising with Inventory Management", In Proceedings of ACM Special Interest Group on E-Commerce (EC00), 1999, pp. 145-149.

[3] D.M. Chickering, D. Heckerman, C. Meek, J.C. Platt, B. Thiesson, "Goal-Oriented Clustering", Technical Report MSR-TR-00-82, Microsoft Corporation, May 2000.

[4] C. Drummnd, R.C. Holte "Exploiting the cost (in)sensitivity of decision tree splitting criteria", In Proceedings of the Seventeenth International Conference on Machine Learning, 2000, pp. 239-246.

[5] C. Elkan, "The Foundations of Cost-Sensitive Learning", In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, 2001.

[6] F. Provost, P. Domingos "Tree Induction for Probability-Based Ranking", Machine Learning, September 2003, pp. 199-215.

[7] J. Stern, Web Metrics: Proven Methods for Measuring Web Site Success, Wiley Publishing Inc., New York, 2002.

[8] J.A. Tomlin "An Entropy Approach to Unintrusive Targeted Advertising on the Web", In Proceedings of the 9th International World Wide Web Conference, 2000.

Figure 1

Product	Impression Ratio Constraint
Product 1	>=.07
Product 2	>=.05
Product 3	>=.06
Product 4	>=.05
Product 5	>=.1
Product 6	>=.04
Product 7	>=.04
Product 8	>=.06
Product 9	>=.04
Product 10	>=.04
Product 11	>=.07
Product 12	>=.04

Figure 2

Product	Impressions	Registrations	Registration Rate
Product 1	1,497,580	2,078	0.139%
Product 2	1,542,919	1,584	0.103%
Product 3	6,329,959	2,121	0.034%
Product 4	1,552,964	2,475	0.159%
Product 5	2,566,991	3,272	0.127%
Product 6	1,532,378	2,177	0.142%
Product 7	7,733,639	4,660	0.060%
Product 8	6,159,841	4,718	0.077%
Product 9	2,063,437	8,633	0.418%
Product 10	5,774,999	2,692	0.047%
Product 11	5,156,031	6,195	0.120%
Product 12	3,012,600	2,246	0.075%
	44,923,338	42,851	

Figure 3

	Training		Validation	
Leaf Node ID	Impressions	Cluster Size	Impressions	Cluster Size
32	1279497	4.1%	548786	4.1%
33	2812817	8.9%	1205408	8.9%
34	863447	2.7%	370196	2.7%
35	2256001	7.2%	967806	7.2%
36	1721255	5.5%	736847	5.5%
37	2511347	8.0%	1076272	8.0%
38	1433647	4.6%	614633	4.6%
39	257067	0.8%	111146	0.8%
40	239450	0.8%	102576	0.8%
41	128621	0.4%	55256	0.4%
42	58826	0.2%	25385	0.2%
43	191812	0.6%	81975	0.6%
44	308061	1.0%	132425	1.0%
45	65734	0.2%	27739	0.2%
46	61519	0.2%	26415	0.2%
47	20355	0.1%	8719	0.1%
48	273835	0.9%	117351	0.9%
49	185227	0.6%	79278	0.6%
50	215917	0.7%	91949	0.7%
51	118679	0.4%	50991	0.4%
52	1230079	3.9%	527031	3.9%
53	95164	0.3%	40723	0.3%
54	96430	0.3%	41452	0.3%
55	24050	0.1%	10331	0.1%
56	4076046	13.0%	1746601	13.0%
57	2483583	7.9%	1065080	7.9%
58	3004160	9.6%	1287944	9.6%
59	762049	2.4%	326533	2.4%
60	1771946	5.6%	760525	5.6%
61	1613843	5.1%	692901	5.1%
62	933219	3.0%	399526	3.0%

Figure 4

Product	Unconstrained Solution	Constrained Solution
Product 1	0.19%	4.0%
Product 2	0.76%	4.0%
Product 3	0.00%	7.0%
Product 4	8.81%	4.0%
Product 5	0.00%	7.0%
Product 6	0.00%	5.0%
Product 7	1.34%	6.0%
Product 8	5.19%	5.0%
Product 9	0.30%	20.4%
Product 10	0.00%	4.0%
Product 11	83.41%	29.6%
Product 12	0.00%	4.0%

Figure 5

Product	Predicted Impression Ratio	Actual Impression Ratio
Product 1	4.0%	4.30%
Product 2	4.0%	3.80%
Product 3	7.0%	7.90%
Product 4	4.0%	3.60%
Product 5	7.0%	5.50%
Product 6	5.0%	7.50%
Product 7	6.0%	5.30%
Product 8	5.0%	2.10%
Product 9	20.4%	22.70%
Product 10	4.0%	3.50%
Product 11	29.6%	28.30%
Product 12	4.0%	5.50%

Autonomous Profit Maximization in Online Search Advertising

Brendan Kitts, Benjamin J. Perry, Benjamin LeBlanc, Parameshvyas Laxminarayan
iProspect, 311 Arsenal Street, Watertown, MA.
bkitts[$\alpha\tau$]*excite.com*

Abstract

We describe an autonomous bidding system that has been used at iProspect for managing on-line paid search advertising spending for 3 years. The system manages hundreds of thousands of on-line advertisements, and is responsible for making decisions, each minute, on millions of dollars worth of assets.

The system uses a tracking system that records cost from customers clicking on advertisements, and revenue from customers purchasing products in real-time. The problem for the bidding system, in its most basic form, is to keep revenue from customers higher than costs from advertising. This is a challenge, since revenue is often characterized by large, infrequent revenue "spikes", where-as cost is incurred every day.

In order to solve this problem, advertisers formally specify their objectives and constraints. For example, an advertiser may specify a goal of producing revenue, and a constraint that for each dollar they spend on customer traffic, they must generate 10 dollars in return from purchases. The bidding agent takes these constraints, and then prices its various advertising assets so as to achieve the advertiser's goal.

Online advertising is a complex and demanding environment. Improperly set bids can result in the equivalent of a home loan being spent in a matter of hours. In order to bid effectively, the system continually creates forecasts of customer behavior, plans out future spending, responds to shocks in the world, and even balances the classical exploration-exploitation trade-off.

The agent's success in solving this problem is evidenced by the financial success of iProspect clients. We present case studies of clients who have used the bidding system. These case studies show that the bidding solution developed by the agent is in some cases orders of magnitude more financially lucrative than the former manual solutions.

Price Optimization in Grocery Stores with Cannibalistic Product Interactions

Brendan Kitts
iProspect, 311 Arsenal Street,
Watertown, MA. 02474.
bkits[$\alpha\tau$]excite.com

Kevin Hetherington
MITRE Corporation
Watertown, MA. 02474.
kevin[$\alpha\tau$]mitre.org

Abstract

We present a SKU-level price optimization system which uses variables that are commonly available to retail Point Of Sales systems. The system is scalable enough to provide optimization and what-if capabilities on 44,791 products. Cross-elasticities such as cannibalistic interactions, are quantified and modeled. The introduction of interactions significantly increases prediction accuracy. We present the results of a pilot study with the system at a grocery retail chain. The retailer implemented price changes on 24 products, and examined the results over 120 days. The price changes resulted in an 29% increase in profit, seasonally adjusted, with increased sales at the category level.

1. Introduction

Retailers have a limited range of options to influence shopper behaviour. Retailers can influence behaviour through four general mediums: (a) price, (b) advertising (eg. weekly newspaper, tv, radio and banner advertising direct mail, and store displays), (c) in-store location (including shelf position, page-hierarchy location, on-page location) and (d) assortment

Of these, price has traditionally been known as one of the easiest and most significant to implement. In a classic study, Lambin (1976) reported that price elasticity is 20 times higher than advertising elasticity (Lambin, 1976). Price changes can be performed with little preparation, and with immediate effects. These advantages contrast with advertising which requires resources to implement. (Simon, 1989).

Selecting prices for 40,000 items in a store is a difficult proposition. The Professional Assignments Group (Pag, 2000) reports that the current practice indicates that retailers might be over-discounting products, with as many as 25%-30% of items being sold at some price discount. With profit margins so low already (1.5% according to the Food Manufacturer's Institute), many retail stores may not be able to sustain aggressive price discounting.

Pricing is made particularly difficult by the fact that products interact with each other. Decreasing the price of one juice item to increase traffic, may merely result in the cannibalization of a more profitable juice brand, as consumers switch from one brand to the other. Similarly, raising prices may have unanticipated negative consequences in unrelated categories, due to a general depression in traffic.

Retailers have long been aware of these price-demand interactions, and have developed various strategies for coping with these effects. One common pricing strategy is the use of "loss leaders". Loss leaders are products which are kept at greatly discounted prices, because they are known to be high-profile, common, and easily comparable between retailers. Typical loss leaders include milk, bread, eggs, and juice.

Loss leaders are presently determined by retailer experience. However, analysis of transaction data should be able to reveal which products are true "loss leaders" and which are not. This can prevent unnecessary discounting. Similarly, not all items need to be kept at discounted prices, and in some cases it makes sense to raise prices.

As a result, a comprehensive approach to pricing - incorporating knowledge of product interactions, consumer demand, and store-wide effects - needs to be developed.

1.1. Outline

Our work differs from previous work in the literature in that we attempt to develop a large-scale, and comprehensive approach to price optimization that includes product interactions. We develop demand models for 40,000 products in the store, and provide this to a global optimization code to optimize price. To validate our system, we ran a large-scale price experiment involving 8 stores, with results collected over 120 days. The results of the experiment were an increase of profit at the item level of 18% unadjusted, or 29% seasonally adjusted, with category profit increases of around 7%.

The outline of this paper is as follows. Section 1 introduces retail data. Section 2 describes the demand model. In section 3, we describe methods for optimizing price using analytical derivatives from the demand model. In section 4, we optimize prices in a retail chain, and examine the results.

2. Retail Data

Let Q be a row vector of quantites sold for each item and $q_i^{(t)}$ be the quantity sold of item i at time t. Let P be a vector of prices, R a vector of revenue, and Π a vector of profit generated from each item on a particular day. T is the number of days of data which we have available. t_0 is today.

Retail data is characterized by a strong 7-day period. The highest volume day of the week is Saturday. This 7 day period can be seen in figure 4. Retailers also experience marked seasonal changes in demand. General Merchandisers experience a dramatic increase in sales during Christmas, shown in figure 3.

The key to optimizing price is to develop an effective model that can predict sales q_i, given any choice of price p_i. However, developing a workable model is extremely challenging.

There are several significant problems making demand forecasting difficult in the retail domain:

(1) Low-volume products. 95% of items in our grocery store's inventory were characterized by low volume. These items have sporadic and discontinuous demand. For example, figure 5 shows a close-up of two items - Newspapers and Funyuns (a kind of snack food). The newspaper timeseries shows a high volume of purchases, with a readily observable 7-day period. Funyuns are purchased only sporadically, and consequently don't show any such pattern.

(2) Heterogeneity of influencing variables. Each product responds to a different set of variables and external factors

(3) Lack of price changes. 70% (31,782 out of 44,791) of items underwent no price change in our grocery data. Many items have the potential to significantly affect demand of other items in the store, but this influence is not seen because those items lay "dormant" in the store until a future time, at which time a price change occurs, and they throw the forecasts. For items that underwent a price change, the most frequent number of changes was 2. over 13.5 months. Because so few prices undergo any price change, for most items it is impossible to optimize their price, since no historical data is available to view the effect of shifting this parameter on store-wide and item sales.

(4) Seasonality: Items behave differently in different seasons.

We have developed techniques to address each of these challenges. We address (1) by transforming the demand series into a smoothed 30-day quantity (section 2.1). We address (2) by performing stepwise regression on a number of variables, and selecting those which are important to each particular item (section 2.2). We address (3) by exploiting cross-price terms (section 2.2.5). We address (4) by introducing time of year variables (section 2.2.2).

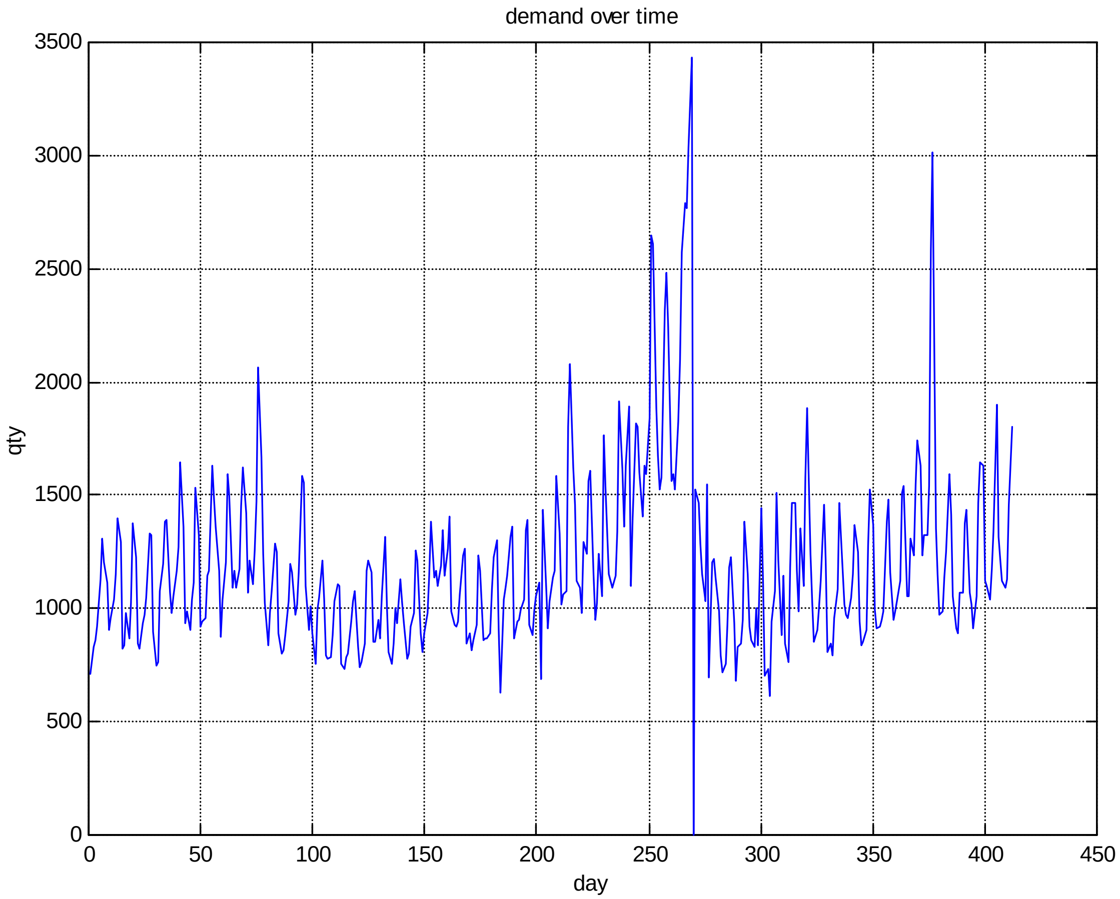


Figure 3: A typical General Merchandiser’s sales over 400 days. The peak is Christmas

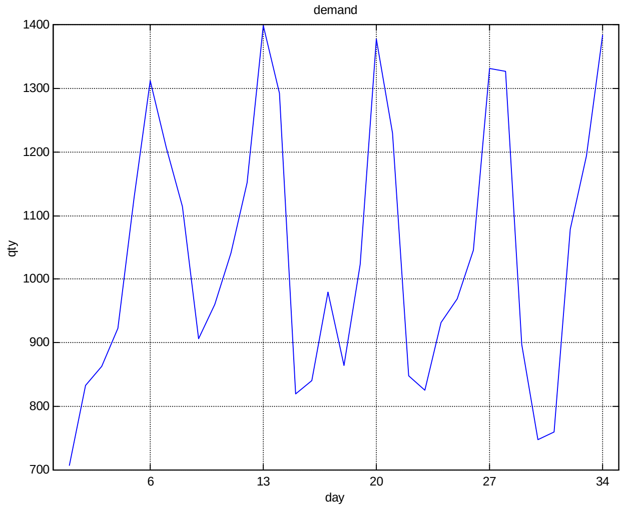


Figure 4: Close-up of the General Merchandiser's demand series (first 35 days). Peaks are 7 days apart.

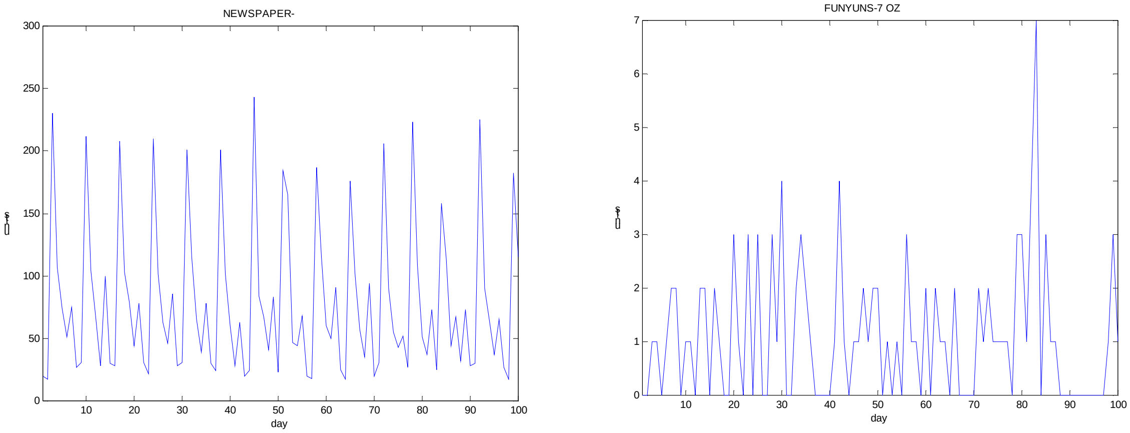


Figure 5: (left) Newspaper sales per day, **(right)** Funyuns sales per day

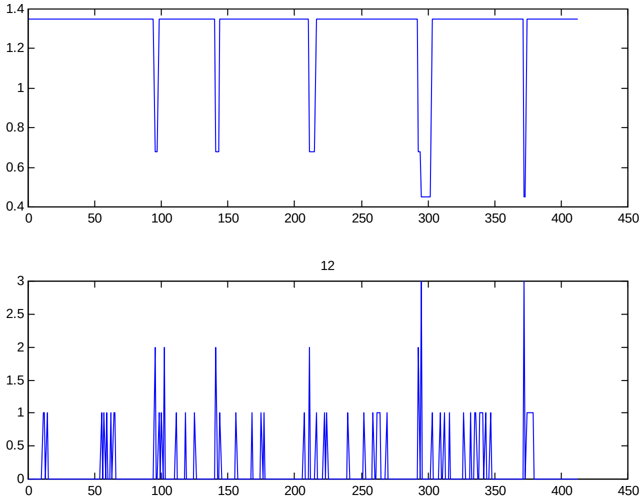


Figure 6: Price and sales are clearly related in this product. The top graphs shows price and the bottom shows units sold. Whenever price drops for a time, there is a small but distinct spike in demand.

2. The Demand Model

2.1. Representation of variables

"Spikey" timeseries can be hard to predict. Let's say that we have a timeseries that consists of just one sale, with the remainder of sales being zero. Our prediction will appear as a single spike on a particular day. Even if the prediction misses the spike by one day, the squared error penalizes the miss. We experimented with many ways to address this problem.

High-frequency components of the timeseries - exactly when in time a sale will occur - are hard to predict. Yet the low-frequency components - the average sales of products purchased over a long period of time - are somewhat easier to predict. Our approach, therefore, was to recode the demand series from amount sold per day, into a "rolling summed amount" over the last week or last month. This transformation accentuates the low-frequency components of the timeseries, at the expense of the high-frequencies.

$$q_i^{(t)} = \sum_{z \in t-W}^{t} q_i^{(z)+} \; ; \; p_i^{(t)} = \sum_{z \in t-W}^{t} p_i^{(z)+}$$

where $p_i^{(t)+} = \begin{cases} p_i^{(t-1)+}, q_i^{(t)+} = 0 \\ p_i^{(t)+}, otherwise \end{cases}$ and W=30

where $q_i^{(t)+}$ is the quantity of item i purchased at time t, and $q_i^{(t)}$ is the sum of the item quantities over the last W days.

The above transformation also results in data which is highly interpretable. Forecasts are now in units of "sales per 30 days". This can be easily used and understood by the retailer for planning and what-if analysis.

Prices also need to be transformed by the same operation. However, and additional problem emerges. Retailers often do not keep good records on when price changes went into effect. Therefore, we picked up those prices from the Point Of Sales data. If a sale did not occur on a particular day, the price of the item is assumed to be the same as the price on the previous day - thus, we "filled in" the price vector on days when we had no sales.

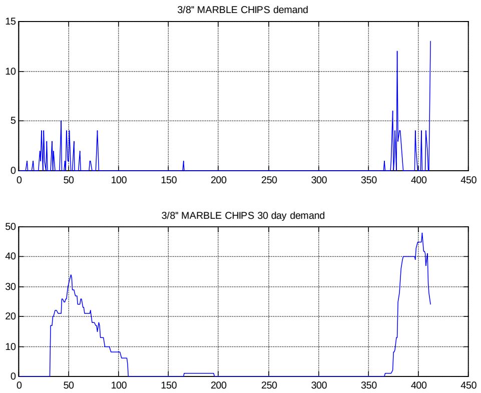


Figure 2: Retail data is characterized by infrequent, sporadic sales. In order to develop workable models, we smooth the data using a moving average window.

2.2. Variables

The demand model was constructed by performing a stepwise regression for each item on a large pool of variables. These variables included:

1. Lag sales: Sales of the item some days in the past.
2. Season indicators: 0-1 variables that indicate a day-of-week or time of the year.
3. Average price: Average price, zscore price, Percent-of-normal price
4. Own price: the price of the item being forecasted
5. Cross-price: the price of other items in the store

2.2.1 Lag Demand. A lag demand term $q_i^{(t0-\Delta t)}$ is the demand Δt days in the past. eg. a lag 5 term is a variable which is the demand 5 days in the past. Because retail data is periodic, lag terms, particularly lag-7 terms, should be extremely predictive.

Although useful, lag-demand terms introduce problems which make them difficult to use operationally. To use demand at lag terms, we would need access to the latest Point Of Sales (POS) data to measure demand, so that we could predict as far ahead as possible. However, buyers operate on planning cycles of up to 2 weeks, which means that only demand terms 2 weeks in the past or more could be used for forecasting, since that's when the decision to buy is made. Another drawback of demand lag is that it is impossible to get demand terms for *future* forecasts. Prices, however, can be planned in advance since they are set by the retailers when planning their promotions. Therefore, future prices can be used in our model, where-as lag-demand data is limited to lag 14 or later.

2.2.2. Day, Month and Season Indicator Variables. A day-of-week indicator vector has 7 elements, with a "1" in the day which is Monday, and a "0" for all other elements. For example, perhaps units sold should be multiplied by a factor of 2 on Saturdays. We encoded days of week and months of year as indicator variables.

2.2.3. Store-wide Sales. When a store undergoes a sale, a large number of product prices drop, resulting in a surge in demand.

We created three storewide price measures - mean price, mean Z-Score price and mean percent of baseline price. The Z-score and percent of baseline metrics were designed to be sensitive to whether the present pricing was higher or lower than normal.

$$f_i^{(t)} = \frac{1}{N}\sum_{i=1}^{N} p_i^{(t)} \ ; \ f_i^{(t)} = \frac{1}{N}\sum_{i=1}^{N} \frac{p_i^{(t)} - \mu(p_i)}{\sigma(p_i)}$$

$$f^{(t)}{}_i = \frac{1}{N}\sum_{i=1}^{N} \frac{p_i^{(t)}}{\mu(p_i)}$$

2.2.4. Own Price. Own-price is the price of an item under consideration. Intuitively, own-price should normally be an important variable for predicting changes in demand. However only 29% of items undergo any own-price change, and this meant that this variable could not be chosen by most of the models!

2.2.5. Cross-price. The use of cross-price variables is rendered challenging by their number. For any given item, there are up to 40,000 other items at the store which could be affecting its sales. This means 40,000 regression equations need to be tested each step of the stepwise procedure!

This problem was solved by creating a full correlation matrix *W'* between the demand series of the target item $q_i^{(t)}$, and the price series for all other 40,000 items, $p_j^{(t)}$ so that for the (*i*,*j*)th element of *W* was:

$$W'(i,j) = \frac{\sum_{t=1}^{T}\left((p_i^{(t)} - \mu(p_i))\cdot(q_j^{(t)} - \mu(q_j))\right)}{\sqrt{\sum_{t=1}^{T}\left((p_i^{(t)} - \mu(p_i))^2 \cdot (q_j^{(t)} - \mu(q_j))^2\right)}}$$

Creating this matrix was comparatively cheap because since a matrix inversion was not required. Yet the correlation coefficient provided a measure of the variance that each variable would account for, were it provided alone to a linear model. We selected the top *n* positive and negative correlations.

This method proved extremely successful. Usually fewer than 50 terms captured the strongest drivers in the store.

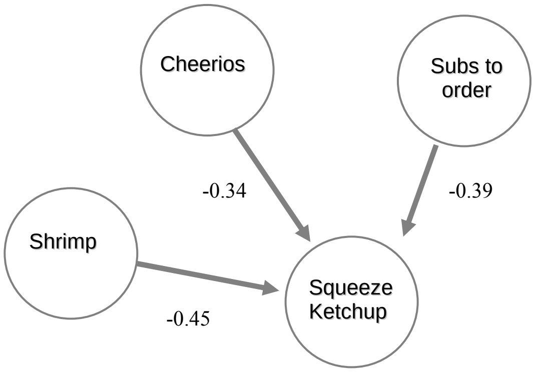


Figure 7: Highest magnitude cross-price w_{ij} terms for Squeeze Ketchup–40 Oz

2.3. Model Form

The model used was a standard regression model with a full matrix of cross-elastic terms, and external variables for seasons, lag-demand, and other variables:

$$Q = P.W + E.X + B$$

$$R = \mathrm{diag}(P.Q.I)$$

$$\Pi = \mathrm{diag}((P\text{-}C).Q.I)$$

I is an identity matrix, diag() returns the on-diagonal elements of a square matrix as a row-vector, *E* is a row vector of external variables including lag-demand, season indicator variables. *X* is a matrix of weights on these external variables. *P* is a row vector of prices for each item, and *W* is a square matrix with weights for each item in the model *i* as a column of weights, and *B* is a row vector of constants. The *W* matrix is a square matrix giving the demand influence (positive or negative) of each item on another item.

W, *B* and *E* are found numerically using least squares procedure described below in the section on "Model Construction".

One might argue that a linear model is a little simplistic. In fact, several functions have been proposed for describing the relationship between price and consumer demand, including linear, multiplicative (exponentially reducing), attraction (sigmoid shaped) and Gutenberg (inverse sigmoid shaped) functions (Simon, 1989).

Simon (1982) and Kucher (1985) noted that all four models provide a reasonably good fit to the observed data. Simon (1989) hypothesized that this might be because all models are approximately linear through their middle range. Non-linearities presumably come into play for extreme prices, but extreme prices are almost never seen in real data. As a result, the linear model has been argued to be an adequate and simple method for modeling the relationship between price and demand.

2.4. Model Construction

The model was trained using the first 230 days as a training set, and the remaining 187 days for testing for each item. A stepwise procedure used an R criterion to determine goodness of fit, and halted if R improvement was less than 0.03.

2.5. Analysis

Cross-price terms were the most commonly selected variables. For sparse items cross-prices were selected 10 times more than lag-demand. For continuous items, cross-price terms were still selected 2.4 times more than lag demand.

Lag terms were selected more frequently as the timeseries became more continuous. Figure 9 shows the difference between using minimum lag of 14 and 5. If lag 5 terms are available, the model uses them as much as the cross-price terms.

However, if these short latency (eg. Lag 5) demand terms are used, then a curious phenomenon occurs. The predicted demand series begins to trail the actual demand series – ie. the forecasts become “late”. This phenomenon does not occur when using longer latency lag terms, such as lag 14 (figure 10 top-left). This means the effect is not caused by a programming error.

Why does the model prefer to be late and simply track the series if short-latency lag terms are available?

For short latency demand series which don’t fluctuate too much, the error resulting from following actual levels is apparently quite small. Although this behavior decreases the error, it may be undesirable. A late forecast provides little comfort to retailers who need to plan inventory levels into the future. In contrast, the lag 14 model both “looks” better (figure 10 right), and predicts level changes when they occur, even it its errors are a higher.

We have also summarized the forecast accuracy of different collections of variables in Table 1. Lag terms score a correlation of R=0.26, but cross-price terms score a correlation of 0.826.

Table 1: Demand model accuracy

Variable Type	R	Mean Absolute Error	Mean Error
Cross-price	0.826	0.28%	-0.14%
Lag-sales per day	0.229	11.49%	9.06%
Lag-sales per 30 days	0.261	8.84%	8.83%
Own price	0.056	57.50%	-30.31%

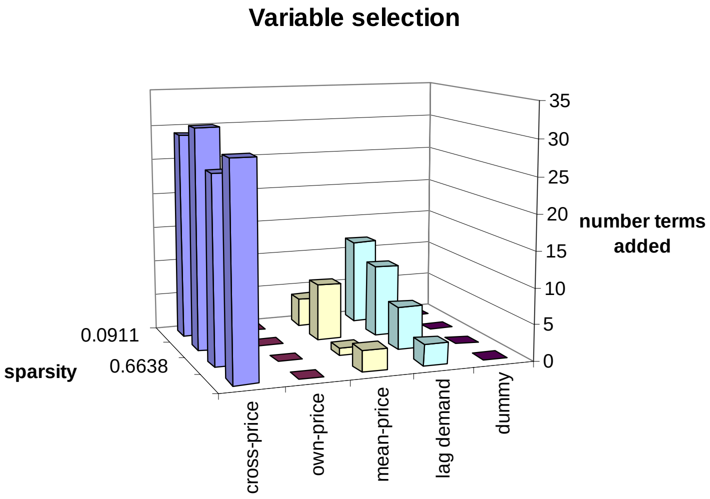


Figure 8: Rate of selection when only lag terms of latency two weeks or higher are allowed into the model. Price is the most favoured variable for determining demand.

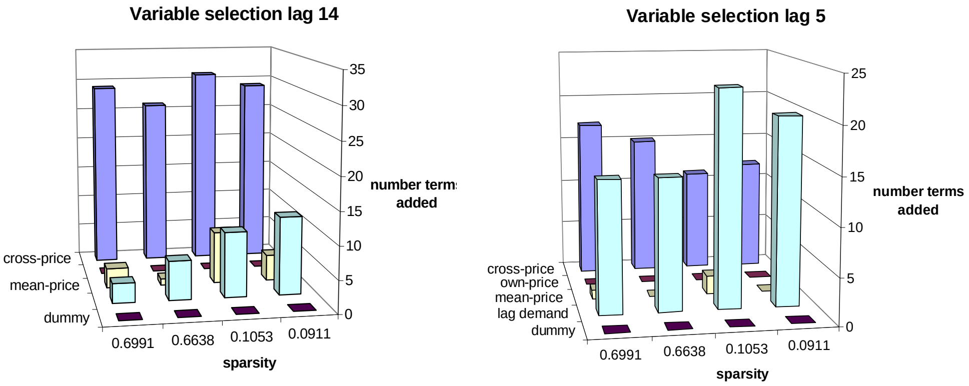
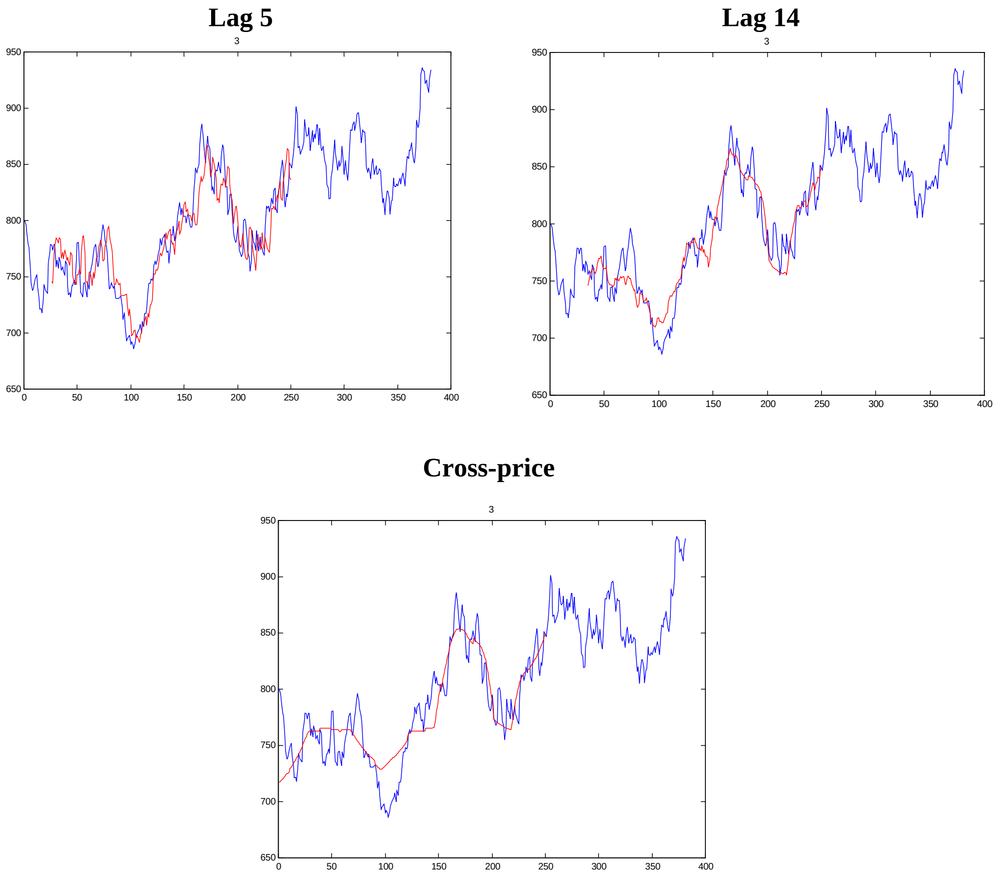


Figure 9: Rate of selection when lag terms with latency 5 days versus 14 days are allowed to be selected. The model places much more emphasis on the lag terms when lag-5 terms are available. It also places more emphasis on the lag terms when the timeseries is more continuous. However, even when lag-5 terms are available, cross-price terms are still highly favored.



Lag 5

Lag 14

Cross-price

Figure 10: Sales forecast versus actual on training set using lag terms of latency 5 or higher **(top left)**, 14 or higher **(top right)**, and cross-price terms only **(bottom)**. If lag 5 terms are allowed, then the model begins tracking the series "late". This kind of behavior minimizes its error, however results in forecasts which arguably may not be able to predict changes. A lag 14 model doesn't show this tracking behaviour, and a model which uses price only also doesn't show this behaviour.

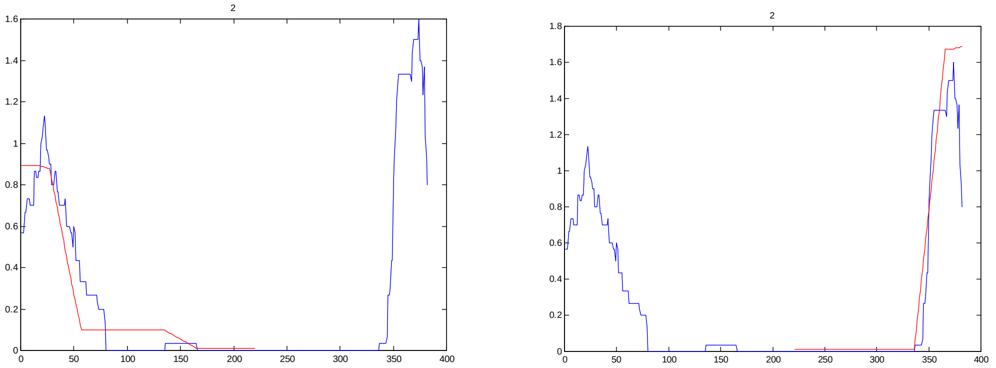


Figure 11: (left) Training set, **(right)** Hold-out set. The vertical axis is the average number of units per day. For slow movers, this average number of items will tend to be less than one. The axis should be multiplied by 30 to give the average number of units over the next 30 days, which

is what the model is designed to predict. The city-block-like patterns are a result of smoothing the demand out over 30 days, which we have found improves prediction markedly on sparse timeseries.

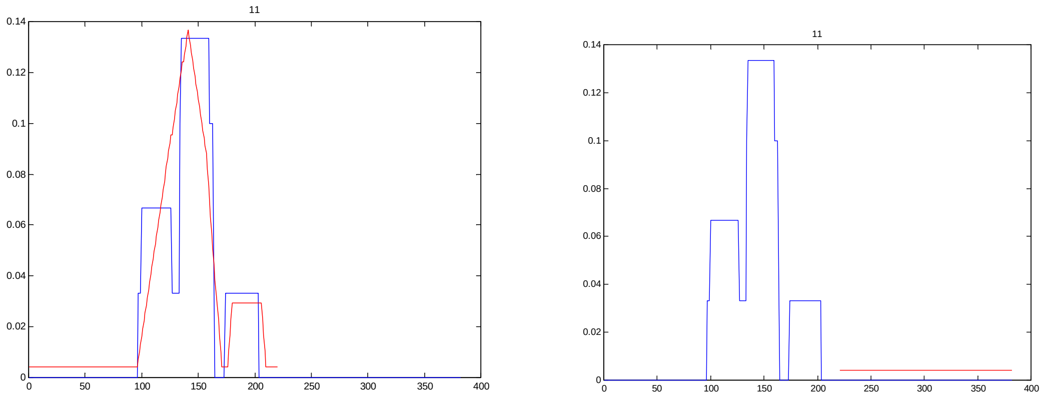


Figure 12: The above model trained on training data **(left)** correctly predicts zero sales on a hold-out period of data of 200 days **(right)**. Note that the predicted level is a little above zero. This is because the model tries to minimize its squared error, and because there have been cases where demand occurred on a non-low-price day, it compromises by elevating its level a little. In terms of cases predicted over the next 200 days, however, this is close to accurate.

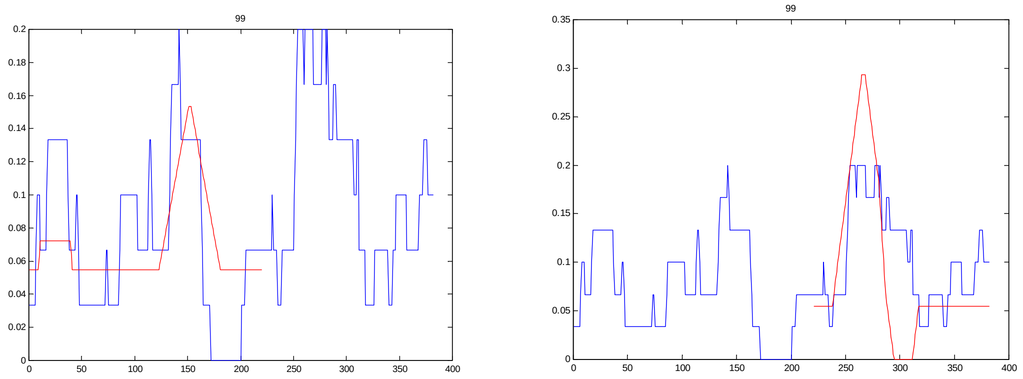


Figure 13: One of the few items in our data that showed a high volume of sales. Performance on hold-out set (a future set of data) is shown at right.

3. Optimization

It is relatively easy to find the optimum price for a single item *i*. Given a profit equation:

$$\pi_i = (p_i w_i + b) \cdot (p_i - c_i)$$

$$\frac{d\pi_i}{dp_i} = 2p_i w_i + b - w_i c_i$$

where π_i is the profit from *i*, p_i the price of *i*, c_i the cost of *i*, and $p_i w_i + b$ is the linear demand equation that estimates the quantity purchased given any price of *i*. An optimum for the profit p_i^* will occur when $\frac{d\pi_i}{dp_i} = 0$. Therefore, we can solve for p_i at this point.

$$p_i^* = \frac{w_i c_i - b}{2w_i}$$

The drawback from using this approach is that this will find the best price of an item without regard to its effect on other items in the store. For example, say that 2 Litre Tropicana Orange Juice sells for $2.50. Single item optimization might predict an optimal profit on Tropicana alone, at a price at $2.60. However, even if were true that $2.60 led to the best profit on Tropicana, this optimum doesn't take into account the store-wide implications of raising price. Perhaps Tropicana Orange juice at the lower price draws in important general sales in the store. In this case, Tropicana might contribute more by actually dropping its price and becoming a consumer drawcard.

The complete demand model which proved to be so accurate in section 1 is used for this multiple, cross-elastic price optimization. The implications of using this model are fairly important. Firstly it is more accurate than the single item-price demand model (Correlation of 5% versus 55%), so the optimization predictions will hopefully be more accurate also. Secondly, the model is completely aware of interactions between a price change on one product, and the impact on up to 40,000 other products in the store. It can conceivably home in on critical items in which price should be dropped very low, for the greater good of the store. This kind of savvy optimization has a lot of potential to find revenue opportunities which human beings mightn't see purely because of the number of items involved, and the fact that interactions occur across category boundaries (something, again, the model has no problems with).

Unfortunately, unlike the single item optimization case, there is no closed form solution to the optimum price vector for the store. Profit optimization becomes a large quadratic programming problem, which must be solved using numerical techniques.

We describe two methods we used below to solve this problem.

Gradient ascent. Gradient ascent uses the first derivative of total profit to guide the algorithm on which direction to move to increase profit. Our derivatives are:

$$\frac{d\Pi}{dP} = Q + (P - C) \cdot W \;;\quad \frac{dR}{dP} = Q + P \cdot W \;;\quad \frac{dQ}{dP} = W^T$$

The simplest possible gradient ascent procedure is to change the vector of prices by a small amount $1 > \propto > 0$ in the direction of the gradient. Repeated application of this update should increase profit. Thus the update procedure to find the optimum for profit would be:

$$P^* = P + \alpha[Q + (P - C) \cdot W]$$

Newton's method. Newton's method obtains the gradient of the first derivative (ie. the second derivative) and extends a line of this gradient to find the point at which this line intercepts 0. Newton's method can be used on non-linear functions, with the understanding that the linearization is an approximation. Since our profit function is quadratic, and the first derivative is linear, Newton's method should exactly find the zero point. The following therefore holds:

$$0 = P - \left(\frac{d^2\Pi}{dP^2}\right)^{-1} \cdot \frac{d\Pi}{dP}$$

Given that we know the second derivatives:

$$\frac{d^2\Pi}{dP^2} = W + W^T \;;\quad \frac{d^2R}{dP^2} = W + W^T \;;\quad \frac{d^2Q}{dP^2} = 0$$

a price which makes the gradient of profit equal to 0 can be found by performing

$$P^* = P - [Q + (P - C) \cdot W] \cdot \left(W + W^T\right)^{-1}$$

4. Experiment

4.1 Design of Experiment

In order to test the effectiveness of the price optimization method, we ran a live experiment. Our participating retailer provided us with 9 test stores in the Mid-West market for conducting trials. We received a feed of 427 days of data in order to conduct the price optimization, and after deploying the prices, we received weekly Point of Sales data from the retailer so that we could monitor our experiment. The data was provided to us as part of an evaluation of a Customer Relationship Management product.

4.1.1 Selection of Items to Optimize. We were not completely free to change prices anywhere in the store, since these might affect existing marketing programs. The retailer identified 7 categories which they were interested in manipulating prices, comprising 430 individual items.

Of those products, we only considered an item a candidate for price optimization if it had undergone at least one price change in the past. Items with no price change carried no information on how their own-price could be adjusted to improve global profit. This made it impossible to use them as an adjustable variable. This constraint ruled out 70% of all UPCs.

We next removed products which had known external dependencies, or had unusual demand curves which suggested external influences. For example, the retailer pointed out that price reductions in the butter category were inappropriate, as prices were directly tied to the price of butter fat, which fluctuated and drove prices at competitor stores as well.

After eliminating these items, we were left with 23/430 = 5% items as our final list of candidates.

4.1.2 Optimization parameters. Our optimization objective was profit. Item cost data was provided by the retailer, and in cases where the data was not available, we used an imputation scheme in which this data was taken from either the category level, or failing that, a 30% margin was assumed. We ran a gradient ascent procedure to determine optimal prices for each of the remaining products.

Price values were constrained to move within their minimum and maximum historical values. if one of the prices increased beyond its upper bound, that price was set to the upper bound, (meaning no movement for that dimension), and the perhaps smaller movement in the other dimensions was allowed, meaning the optimization slid along the boundary in its allowable directions.

The final price changes for each of these items is shown in Table 2. The price changes consisted of 12 decreases and 14 increases. The average price change was –\$0.05.

After delivering these prices to the retailer, the price experiment was performed over 120 days, with a Point of Sales feed collected from the retailer each week and analyzed.

4.1.3 Control Groups. We were provided with three control groups to measure our results.

For each product, the retailer provided us with two stores that would act as controls and 2 stores which would receive the price changes.

The retailer also ran a separate test group which consisted of price changes that the store managers made on a small number of products. The exact price changes were not known to us at the time, and they were implemented simultaneously with our new prices. These retailer price changes are listed in Table 8.

Finally, we were able to use the historical behavior of the products that we changed as a way of quantifying the impact of the price intervention.

4.4.4 Measurement of results. Customer loyalty card numbers allowed us to measure changes in customer behavior including visits to the store. However, although we used this data to analyze the impact of our program, the price optimization algorithm we have described works on completely anonymous, non-loyalty-card, Point Of Sales scanner data. We were able to measure the impact of the intervention on six variables: quantity purchased, revenue, profit, visits and number of distinct customers and distinct baskets.

Table 2: Optimal Price changes

Item description	Intervention type	Price before	Price after	Change
IGA CUT ASPARAGUS-14.5 OZ	increase	1.19	1.29	0.10
IGA 3 SV CUT GREEN BEANS-8 OZ	increase	0.39	0.43	0.04
IGA CUT GREEN BEANS-14.5 OZ	increase	0.49	0.59	0.10
IGA FRENCH STYLE GRN BEAN-14.5	increase	0.49	0.59	0.10
IGA FR STY GREEN BEANS-8 OZ	increase	0.39	0.43	0.04
IGA DICED CARROTS-14.5OZ	increase	0.49	0.59	0.10
IGA MEDIUM SLICED CARROTS-14.5	increase	0.49	0.59	0.10
IGA CREAM STYLE CORN-14.5 OZ	increase	0.49	0.59	0.10
IGA CREAM STYLE CORN-8 OZ	increase	0.39	0.43	0.04
IGA WHOLE KERNEL CORN-15.25	increase	0.49	0.59	0.10
IGA WHOLE KERNEL CORN-8.0 OZ	Increase	0.39	0.43	0.04
IGA MIXED SWT PEAS-15 OZ	Drop	0.59	0.49	-0.10
IGA SLICED POTATOES-15 OZ	increase	0.59	0.67	0.08
IGA WHOLE POTATOES-15 OZ.	increase	0.59	0.67	0.08
SKIP CHUNK PEANUT BUTTER-18 OZ	Drop	2.19	2.05	-0.14
SKIP CRMY PEANUT BUTTER-18 OZ	Drop	2.19	2.05	-0.14
SKIPPY R.FAT CHUNKY P.BUTTER-1	Drop	2.19	2.05	-0.14
SKIPPY R.FAT CREAMY P.BUTTER-1	Drop	2.19	2.05	-0.14
TROP PURE PREM ORANGE JCE-64 O	Drop	3.39	3.09	-0.30
TROP PURE PREM HOMESTYLE-64 OZ	Drop	3.39	3.09	-0.30
TROP PURE PREM GROVESTAND-64 O	Drop	3.39	3.09	-0.30
TRP PURE PREM + CALCIUM-64 OZ	Drop	3.39	3.09	-0.30
			Average	**-0.04**

Table 3: Item-by-item summary of profit change due to intervention

descript	intervention type	profbefore control	profbeforestd control	profduring control	profafterstd control	profbefore exp	profbeforestd exp	profduring exp	profafterstd exp	%profchange	%controlchange	absolute diff
G MILLS CIN TOAST CRUNCH-14 OZ	drop	2.63	5.95	1.66	2.71	3.33	6.80	0.65	0.70	-80%	-37%	-1.71
KEL CRISPIX-12 OZ	drop	1.36	1.16	0.96	0.80	1.12	0.97	0.52	0.49	-54%	-29%	-0.20
KEL FST MINI WHEATS-24.3 OZ	increase	0.83	0.94	1.08	0.94	1.37	0.96	1.82	1.45	33%	30%	0.20
KELLOGG RICE KRISPIES-13.5 OZ	drop	1.68	1.27	1.64	1.47	2.11	2.24	1.37	1.19	-35%	-2%	-0.71
IGA CUT ASPARAGUS-14.5 OZ	increase	0.50	0.50	0.43	0.49	0.37	0.45	0.53	0.68	45%	-13%	0.23
IGA 3 SV CUT GREEN BEANS-8 OZ	increase	0.13	0.15	0.11	0.13	0.09	0.12	0.14	0.17	53%	-14%	0.07
IGA CUT GREEN BEANS-14.5 OZ	increase	1.81	3.25	2.07	2.88	1.85	2.49	2.90	3.78	56%	15%	0.78
IGA FRENCH STYLE GRN BEAN-14.5	increase	2.31	3.00	1.20	1.70	1.57	1.88	1.65	1.24	5%	-48%	1.19
IGA FR STY GREEN BEANS-8 OZ	increase	0.06	0.11	0.07	0.11	0.06	0.10	0.09	0.14	43%	7%	0.02
IGA DICED CARROTS-14.5OZ	increase	0.18	0.23	0.16	0.22	0.17	0.23	0.23	0.36	37%	-6%	0.07
IGA MEDIUM SLICED CARROTS-14.5	increase	0.43	0.31	0.40	0.37	0.40	0.31	0.63	0.53	55%	-6%	0.25
IGA CREAM STYLE CORN-14.5 OZ	increase	1.52	0.99	0.81	0.61	1.25	1.15	1.28	0.90	2%	-47%	0.73
IGA CREAM STYLE CORN-8 OZ	increase	0.13	0.20	0.16	0.20	0.20	0.25	0.18	0.23	-12%	25%	-0.06
IGA WHOLE KERNEL CORN-15.25	increase	3.04	1.66	1.68	1.09	2.62	2.16	3.16	1.69	21%	-45%	1.90
IGA WHOLE KERNEL CORN-8.0 OZ	increase	0.18	0.17	0.15	0.15	0.13	0.15	0.18	0.19	42%	-17%	0.08
IGA MIXED SWT PEAS-15 OZ	drop	1.16	0.90	0.47	0.44	1.53	1.27	0.95	0.66	-38%	-59%	0.10
IGA SLICED POTATOES-15 OZ	increase	0.29	0.36	0.33	0.44	0.23	0.32	0.24	0.35	3%	14%	-0.03
IGA WHOLE POTATOES-15 OZ.	increase	0.33	0.36	0.44	0.60	0.30	0.45	0.28	0.44	-4%	36%	-0.13
SKIP CHUNK PEANUT BUTTER-18 OZ	drop	0.31	0.33	0.32	0.38	0.30	0.37	0.23	0.23	-22%	4%	-0.08
SKIP CRMY PEANUT BUTTER-18 OZ	drop	1.11	0.74	1.10	0.70	0.94	0.72	0.61	0.45	-34%	-1%	-0.31
SKIPPY R.FAT CHUNKY P.BUTTER-1	drop	0.18	0.30	0.21	0.27	0.20	0.30	0.12	0.17	-41%	15%	-0.11
SKIPPY R.FAT CREAMY P.BUTTER-1	drop	0.34	0.38	0.38	0.43	0.39	0.36	0.24	0.25	-38%	12%	-0.19
TROP PURE PREM ORANGE JCE-64 O	drop	2.74	2.42	2.90	2.57	2.31	1.96	2.54	2.33	10%	6%	0.06
TROP PURE PREM HOMESTYLE-64 OZ	drop	1.69	1.73	1.96	1.96	1.79	1.61	2.69	2.41	50%	16%	0.61
TROP PURE PREM GROVESTAND-64 O	drop	2.24	1.73	2.25	2.00	1.79	1.65	2.62	2.09	46%	0%	0.81
TRP PURE PREM + CALCIUM-64 OZ	drop	2.74	1.97	2.83	2.43	1.97	1.92	2.43	2.34	23%	3%	0.36
Total	**All**	**1.15**		**0.99**		**1.09**		**1.09**		**6%**	**-5%**	**0.15**

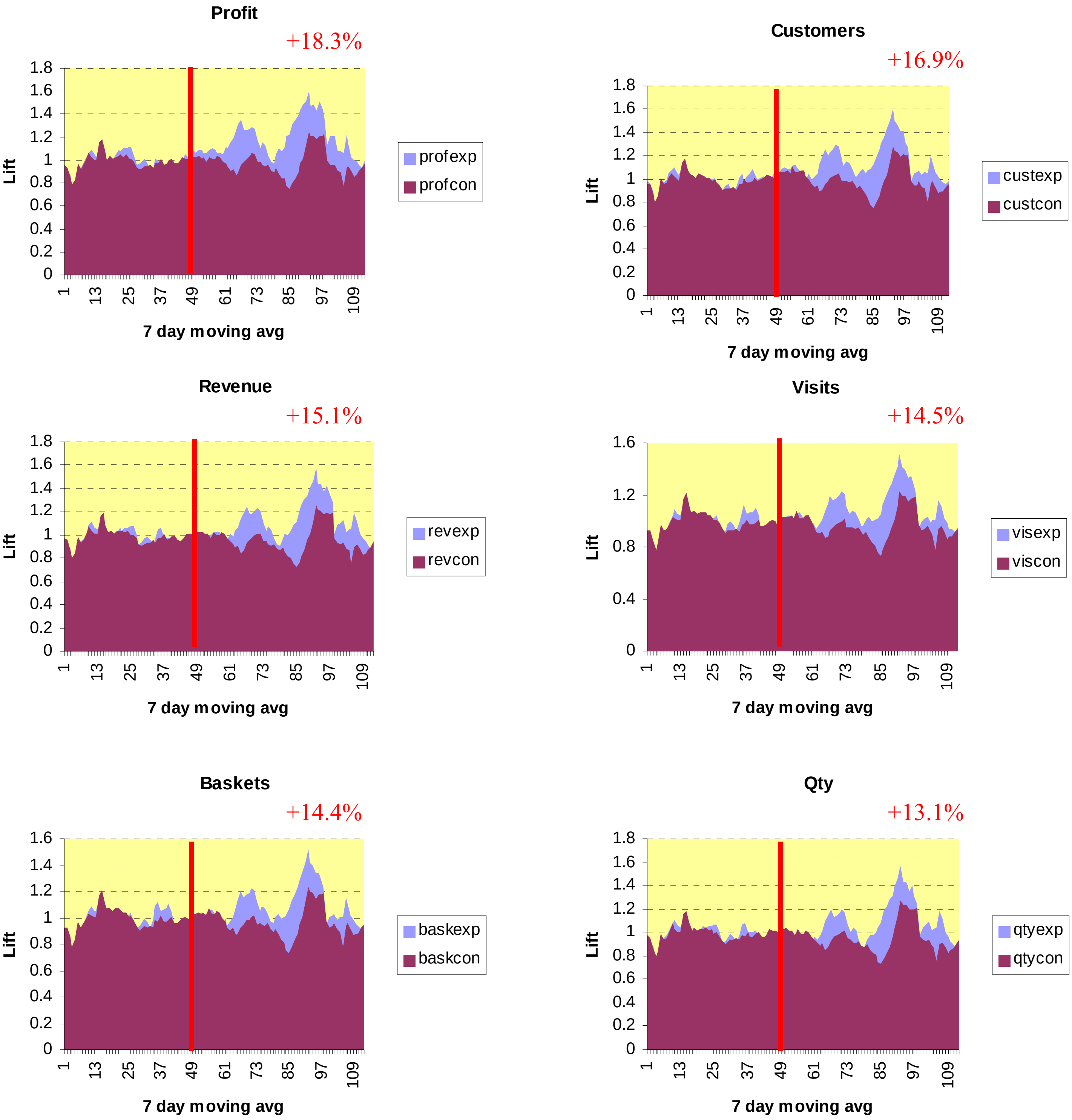


Figure 15. Overall results as a timeseries

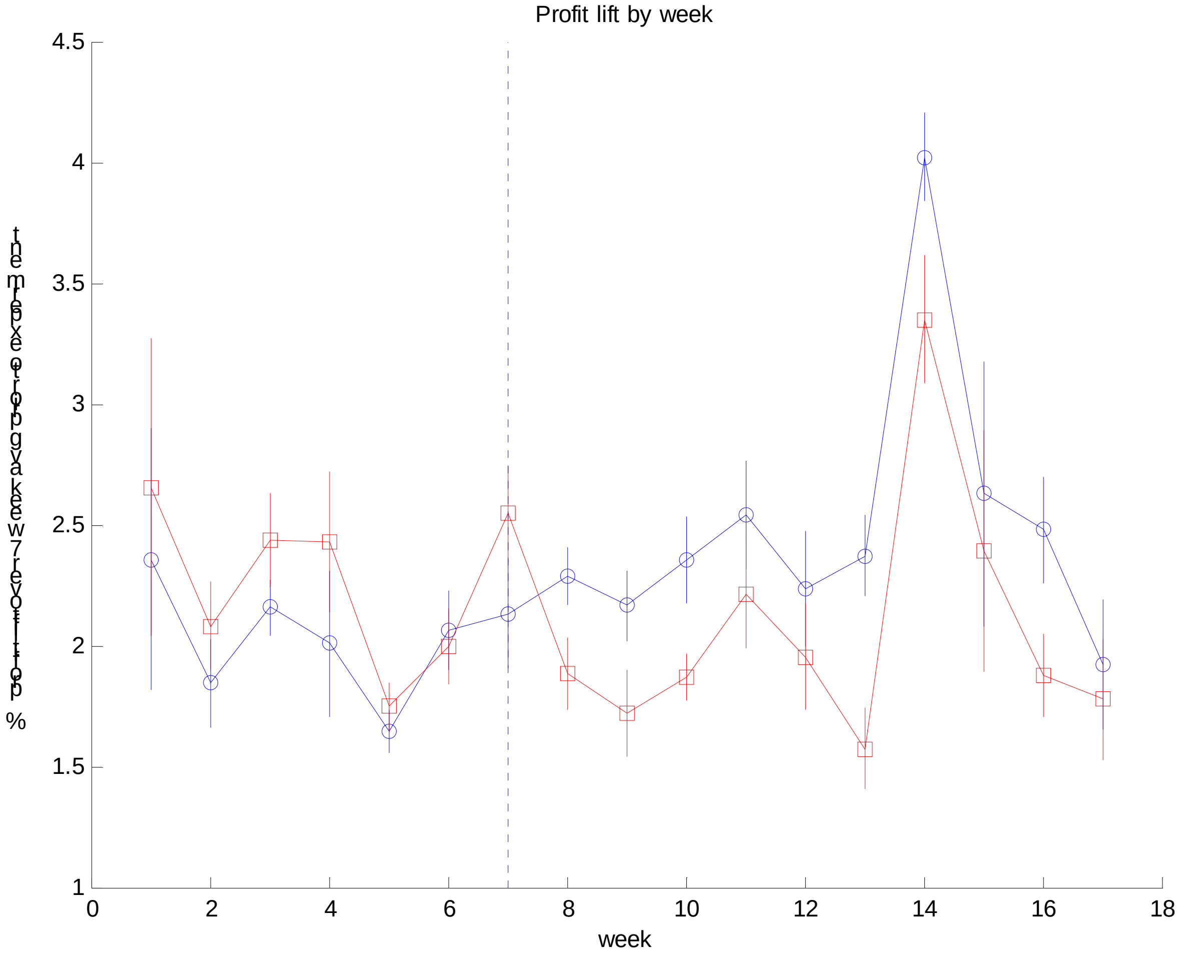


Figure 16: Profit per week at experimental versus control stores, for profit of affected items.

5. Results

5.1 Lift

Lift quantifies the increase in a customer measure, measured against the score prior to the intervention. To calculate lift we divided the measure after the intervention by the measure before the intervention.

$$lift_G = \frac{\mu(R_G{}^{t+1})}{\mu(R_G{}^{t})} - 1$$

where $\mu(R_G{}^t)$ is the average of metric R for group G at time t.

By this measure, profit rose by an average of 18% per product in the experimental group, and decreased by an average of 13% per product in the control.

5.2 Seasonally Adjusted Lift

Because we measure quantities through time, various external influences may be occurring during the measurement period. These external influences, such as seasonality, should ideally be factored out when reporting lifts. For instance, if juice increased by 10% in a control group, and 12% in an experimental group, we should normally assume that the juice would have experienced a 10% increase in either group. Therefore, the actual lift for the experimental group might be only 12% - 10% = 2%.

Seasonally adjusted lift is therefore the experimental lift minus the control lift. This results in the number of percentage points higher that the experimental group was compared to the control group.

$$Lift_G{}' = \frac{\mu(R_G{}^{t+1})}{\mu(R_G{}^{t})} - \frac{\mu(R_C{}^{t+1})}{\mu(R_C{}^{t})}$$

In terms of simple before-after lift scores in the control group, *all* customer measures decreased during the experiment. Profit, for example, declined by 13%. Absolute profit in the experimental group, measured as before versus after, bucked the trend and increased by 17%.

Both the experimental and control groups underwent declines during the experimental period, as part of a seasonal trend. Therefore, "seasonally adjusted" lift scores can be used to measure the improvement in the experimental group after accounting for that general decline.

According to this score, quantity increased 12%, baskets 15%, visits 15%, customers 19%, revenue 24% in the experimental group. Profit - which also increased by 18% in real terms - increased 29% in seasonally adjusted terms.

All variable changes were statistically significant according to t-test and Wilcoxon rank sign test at the p<0.01 level.

Table 4: Overall Results - Seasonally Adjusted Lift in Six Metrics

Measure	Seasonally Adjusted Lift
revenue	23.54%
profit	29.37%
customers	19.50%
visits	15.47%
baskets	15.52%
quantity	12.12%

Table 5: Change in profitability of items selected for price optimization, but which were not changed (Control Stores)

Measure	Before Control	After Control	Change Control	Control Before Std	Control After Std
revenue	$3.69	$3.17	-14%	3.23	3.00
profit	$1.06	$0.93	-13%	0.99	0.92
customers	2.56	2.04	-21%	2.13	1.85
visits	2.76	2.07	-25%	2.21	1.90
baskets	2.75	2.06	-25%	2.19	1.89
quantity	4.47	3.07	-31%	3.96	3.04

Table 6: Change in profitability of Items with price changes in Experimental Stores

Measure	Before Exp	After Exp	Change Exp	Exp Before Std	Exp After Std
revenue	3.14	3.44	+9%	3.05	2.99
profit	0.93	1.09	+17%	0.92	0.98
customers	2.06	2.04	-1%	1.97	1.62
visits	2.29	2.07	-10%	2.29	2.07
baskets	2.29	2.07	-9%	2.06	1.65
quantity	3.73	3.02	-19%	3.73	2.58

5.3 Cannibalization

Our technique for quantifying cannibalization was to examine the profitability of categories in which products were either changed or held the same. If the profitability of the category stayed the same or decreased compared to the same category in control stores, then cannibalization could have occurred.

In all categories except one (Peanut butter) profit increased. Therefore we will conclude that for 5 of 6 categories, no cannibalization was apparent. Overall the contribution of the experiment to global store profitability seems to have been overwhelmingly positive.

Table 7: Presence of Cannibalization

Category	Control % Change	Exp % Change	Excess Lift
Canned Corn	-25%	-17%	8%
Canned Beans	-21%	-16%	5%
Canned Vegetables	-16%	-11%	5%
Canned Peas	-21%	-10%	11%
Peanut butter	-7%	-5%	-2%
Refrig. Juice	10%	-7%	3%

5.4 Control Price Changes

The retailer implemented changes resulted in universal drops in profitability, revenue, and sales.

Table 8: Price changes implemented by retailer

Item	Price before	Price after	Change
G Mills Cin Toast Crunch 14 Oz	3.49	3.23	-0.26
Kel Crispix 12 Oz	3.19	2.98	-0.21
Kel Fst Mini Wheats 24.3 Oz	3.39	3.69	0.30
Kellogg Rice Krispies 13.5 Oz	3.49	3.19	-0.30
Total			**-0.27**

Table 9: Impact of retailer price changes

Measure	Change
Profit	-24.45%
Revenue	-18.01%
Visits	-16.45%
Quantity	-16.44%
Baskets	-16.53%
Customers	-9.39%

6. Conclusion

We have presented a model for optimizing price at grocery stores which incorporates knowledge of cross-elastic price interactions between products. The model is simple and can be implemented with anonymous Point Of Sales scanner data. The use of Point Of Sales data only is useful, because records of price changes, promotions, sales, are often kept in non-standardized data formats - or are not kept at all in electronic format. In contrast, Point Of Sales data sources are highly standardized and can be easily utilized.

The model has been used to create a "what-if analysis" workbench, where anticipated prices can be tested for their impact on global store profit, traffic, as well as cross-effects on other items. This can provide much insight into what might happen if a particular planned price is implemented.

The model was tested at a retail chain, and shown to produce an impressive increase in profit of 29% seasonally adjusted, and 18% in real terms.

Although these increases are excellent, only 29% of items had undergone a price change, and only 5% were found to be suitable candidates for implementing the price recommendations. As a result, some degree of human validation of prices should be undertaken before implementing the prices in the store. We took this approach in this experiment, and generated excellent results.

Although price optimization has been our focus, it was only one of several applications that were prototyped during our study. The model can be used to determine the outcome of any planned price promotion, list the strongest drivers of traffic to the store, which products cannibalize each other, and what products have long-range, cross-category effects on other products at the store. These analyses could be provided to support planning and analysis by human pricing managers trying to get the most out of their store categories.

7. References

[1] E. Kucher, *Scannerdaten und Preissensitivitat bei Konsumgutern,* Gabler-Verlag: Wiesbaden, 1985.

[2] J. Lambin, *Advertising, Competition and Market conduct in Oligopoly Over Time*, North-Holland Publishing Company, Amsterdam, 1976.

[3] H. Simon, *Price Management*, North-Holland Publishing Company, Amsterdam, 1989.

[4] Professional Assignments Group, www.pag.com.au, 2000

[5] E. Watson, Retailers feel the heat as margins slump to new lows, *Food Manufacture website*, http://www.foodmanufacture.co.uk/news/fullstory.php/aid/2235/Retailers_feel_the_heat_as_margins_slump_to_new_lows.html, 6th October, 2005.

Market Basket Recommendations for the HP SMB Store

Pramod Singh
Hewlett-Packard Company
14231 Tandem Blvd
Austin, TX - 78727
(512) 432 8794
pramod.singh[ατ]*hp.com*

A Charles Thomas
Hewlett-Packard Company
20555 Tomball Parkway
Houston, TX - 77070
(281) 514 1804
cthomas[ατ]*hp.com*

Ariel Sepulveda
Hewlett-Packard Company
PO Box 4048
Aguadilla, PR - 00605
(787) 819.6057
ariel.sepulveda[ατ]*hp.com*

Abstract

The Analytics team at Hewlett-Packard recently executed a manually-driven cross-sell/up-sell pilot in the Small and Medium Business online store and call center. The pilot, for which management dictated a 1 month development timeframe, utilized sales transaction, product configuration, and product availability data. Leveraging Market Basket analysis techniques among a small subset of available product SKUs, the pilot yielded a ROI of more than $300K/month and more importantly, gave birth to greater opportunities to further showcase the power of analytics and data driven decision-making at HP.

1. Introduction

For the purpose of enhancing direct sales transaction revenue and profit, the Analytics team at HP was asked to execute a cross-sell/up-sell project for the Small and Medium Business (SMB) store's online site and call center. Cross-selling includes, among others, adding a monitor, docking station, or a digital camera to a notebook purchase. An up-sell is loosely defined as "inside the box." For example, up-selling would include adding anything that enhances value of a PC, such as upgraded memory, hard drive, or a DVD drive.

The pilot's overall goal was to increase the revenue and margin of the store by increasing average order value (AOV) and attach rate per product by implementing an analytic solution. This pilot served as a proof of concept, which may translate into future investment in a more automated alternative. To create the manual execution, a process was built where existing sales, product information, and internal marketing information were integrated and analyzed to identify potential cross-sell and up-sell recommendations. Note that the SMB store is public. In other words, the team could not link visitors to their historical purchase behavior. Thus, they leveraged techniques more geared towards aggregate-level analyses.

The pilot was originally designed to generate learning that would lead to a staged regional implementation and eventual world-wide program which enables segment marketing and telesales teams to increase attach, margin and revenue in the direct business by providing the foundation, tools and capabilities to enable the generation and delivery of analytically driven offers.

Recommendations for more than 25 models of desktops, notebooks, printers, servers, storage and workstations product categories, representing more than 100 SKUs, were generated during the pilot. Maximization of revenue coverage and product availability were key criteria in the SKU selection process. The recommended offers were rendered at the product configurator (web page where customers can customize their machine) and the check-out pages of the selected products.

The 5 person team (3 from Analytics and 2 from IT) assigned to the project planned two incremental releases. Release 1 leveraged historical purchase data to generate recommendations based upon what has traditionally been purchased together in the past. The objective of this phase was to maximize expected cross-sell and up-sell revenue. Release 2 added product profitability data to historical purchase data to yield recommendations based upon traditional product affinities and those products most profitable to the company. In this phase, the team planned to maximize a weighted function of revenue and profitability, weights being provided by the business owners.

Overall, the project required data preparation, statistical analysis, business review and final recommendation generation, recommendation execution, periodic reporting, and post-pilot performance evaluation.

2. Data Preparation

For meeting the objectives of using margin and revenue in the scoring function to rank available recommendations, and also to provide lead time as a potential criterion, different HP data sources, all new to the analytics and supporting IT team were identified, evaluated and integrated. Sources included sales transactional data, product configuration, product hierarchy, and product availability data among others. Data challenges included a high percentage of missing values in some key fields and difficulty associating the appropriate price for a SKU given frequent price changes. In cases where these values were missing, the analysts developed a hierarchical imputation process in which missing margins for SKUs would be estimated based on similar SKUs in the product hierarchy.

Once all the data were available, a flat table was created with dimensions of about 3,000,000×20 for more than a year of data. Some of the most significant fields in that table were Order ID, Date, SKU, SKU Description (descriptions at different levels of the product hierarchy), Quantity, Price, Cost, and Lead Time (an indicator for product availability). The winning Cross-sell recommendations were selected from thousands of competing SKUs. A sample of the data is shown in Table 1. In this table, the bold faced and shaded rows describe base units or machine sold with upgrades described in the remaining bold face rows.

The next section describes in more detail how the team analyzed the data to yield the recommendations.

3. Data Analysis

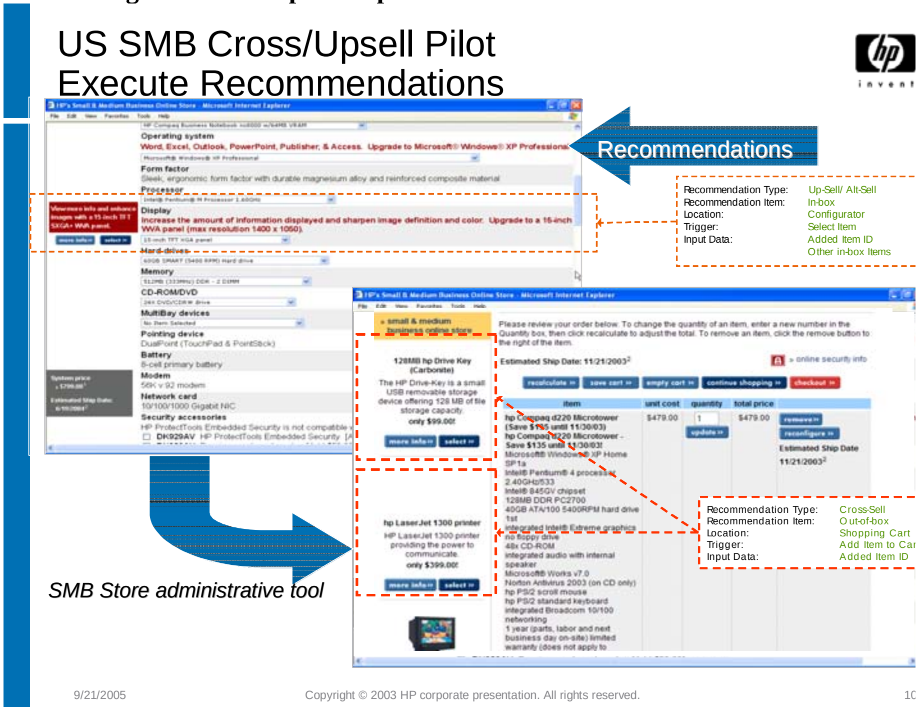
Before merely jumping in and performing the analysis, the team decided to operationalize the dependent variables: cross-sell and up-sell. Cross-sell refers to "outside-the-box" purchases that customers add to a base unit. One simple example is the purchase of a printer cable, print cartridges, or paper along with a printer. On the other hand, up-sell refers to upgrades to systems that are sold as a box. For example, a desktop bundle including monitor, software, and all the internal features like DVD writer or memory is considered to be a "box." Therefore, upgrading from a 256 MB memory to 512 MB memory in such a desktop package is considered to be an "up-sell." However, in cases where a customer does not select a package, but rather chooses to purchase a desktop alone, selling a 21 inch monitor in many cases may also be considered as a cross-sell because the CPU might not include a monitor as part of the main product or box.

Due to the different options for configuring a desktop, up-sell becomes an important characteristic of customer behavior. For configuration purposes, up-sells are offered right in the configuration page (where customers customize the box to meet their needs), whereas cross-sells can be offered on the configurator as well as in the cart page (see Figure 1 below as implemented on the www.smb.compaq.com website).

Table 1: Sample of fields for order 849608

tie_group_id	parent_comp_id	comp_id	osku	description	qty
1	6706	6706	301897-B22	HP StorageWorks MSL5030 Tape Library, 1 LTO Ultri	1
2	0	22814	C7971A	HP Ultrium 200 GB Data Cartridge	10
3	0	22961	C7978A	HP Ultrium Universal Cleaning Cartridge	1
4	**7167**	**7167**	**292887-001**	**Intel Xeon Processor 2.40GHz/512KB**	**2**
4	**7167**	**26919**	**1GBDDR-1BK**	**1GB Base Memory (2x512)**	**2**
4	**7167**	**20481**	**286714-B22**	**72.8 GB Pluggable Ultra320 SCSI 10,000 rpm Univ**	**2**
4	**7167**	**20478**	**286713-B22**	**36.4GB Pluggable Ultra320 SCSI 10,000 rpm Unive**	**2**
4	**7167**	**24578**	**326057-B21**	**Windows® Server 2003 Standard Edition + 5 CALs**	**2**
5	7167	7167	292887-001	Intel® Xeon Processor 2.40GHz/512KB	4
5	7167	20478	286713-B22	36.4GB Pluggable Ultra320 SCSI 10,000 rpm Univers	4
5	7167	20478	286713-B22	36.4GB Pluggable Ultra320 SCSI 10,000 rpm Univers	4
5	7167	24578	326057-B21	Windows® Server 2003 Standard Edition + 5 CALs	4
6	7167	7167	292887-001	Intel® Xeon Processor 2.40GHz/512KB	1
6	7167	26919	1GBDDR-1BK	1GB Base Memory (2x512)	1
6	7167	20484	286716-B22	146.8 GB Pluggable Ultra320 SCSI 10,000 rpm Unive	1
6	7167	20484	286716-B22	146.8 GB Pluggable Ultra320 SCSI 10,000 rpm Unive	1
6	7167	24578	326057-B21	Windows® Server 2003 Standard Edition + 5 CALs	1
7	8947	8947	FA107A#8ZQ	iPAQ h5550 Pocket PC	2
8	0	0	FA136A#AC3	hp 256MB SD memory	2
9	0	0	FA121A#AC3	hp iPAQ Compact Flash Expansion Pack Plus	2
10	0	0	271383-B21	Compaq 320MB flash memory card	1

Figure 1 – Example of Up-sell and Cross-sell recommendations



US SMB Cross/Upsell Pilot
Execute Recommendations

invent
HP's Small & Medium Business Online Store - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Recommendations
more info » select »
recalculate » save cart » empty cart » continue shopping » checkout »
update » remove » reconfigure »
» online security info
» small & medium business online store

HP Compaq Business Notebook nx8000 w/64MB VRAM

Operating system
Word, Excel, Outlook, PowerPoint, Publisher, & Access. Upgrade to Microsoft® Windows® XP Professional
Microsoft® Windows® XP Professional

Form factor
Sleek, ergonomic form factor with durable magnesium alloy and reinforced composite material

Processor
Intel® Pentium® M Processor 1.60GHz

View more info and enhance images with a 15-inch TFT SXGA+ WVA panel.

Display
Increase the amount of information displayed and sharpen image definition and color. Upgrade to a 15-inch WVA panel (max resolution 1400 x 1050).
15-inch TFT XGA panel

Hard drives
60GB SMART (5400 RPM) Hard drive

Memory
512MB (333MHz) DDR - 2 DIMM

CD-ROM/DVD
24X DVD/CDRW drive

MultiBay devices
No Item Selected

Pointing device
DualPoint (TouchPad & PointStick)

Battery
8-cell primary battery

Modem
56K v.92 modem

Network card
10/100/1000 Gigabit NIC

Security accessories
HP ProtectTools Embedded Security is not compatible w
☐ DK929AV HP ProtectTools Embedded Security [A

System price
» $799.00[1]

Estimated Ship Date:
6/10/2004[2]

Recommendation Type: Up-Sell/ Alt-Sell
Recommendation Item: In-box
Location: Configurator
Trigger: Select Item
Input Data: Added Item ID
Other in-box Items

Please review your order below. To change the quantity of an item, enter a new number in the Quantity box, then click recalculate to adjust the total. To remove an item, click the remove button to the right of the item.

Estimated Ship Date: 11/21/2003[2]

128MB hp Drive Key (Carbonite)
The HP Drive-Key is a small USB removable storage device offering 128 MB of file storage capacity.
only $99.00!

hp LaserJet 1300 printer
HP LaserJet 1300 printer providing the power to communicate.
only $399.00!

item	unit cost	quantity	total price	
hp Compaq d220 Microtower (Save $135 until 11/30/03) hp Compaq d220 Microtower - Save $135 until 11/30/03! Microsoft® Windows® XP Home SP1a Intel® Pentium® 4 processor 2.40GHz/533 Intel® 845GV chipset 128MB DDR PC2700 40GB ATA/100 5400RPM hard drive 1st integrated Intel® Extreme graphics no floppy drive 48x CD-ROM integrated audio with internal speaker Microsoft® Works v7.0 Norton Antivirus 2003 (on CD only) hp PS/2 scroll mouse hp PS/2 standard keyboard integrated Broadcom 10/100 networking 1 year (parts, labor and next business day on-site) limited warranty (does not apply to	$479.00	1	$479.00	Estimated Ship Date 11/21/2003[2]

Recommendation Type: Cross-Sell
Recommendation Item: Out-of-box
Location: Shopping Cart
Trigger: Add Item to Car
Input Data: Added Item ID

SMB Store administrative tool

9/21/2005 Copyright © 2003 HP corporate presentation. All rights reserved. 1C

Up-sells are also challenging in the sense that compatibility must be taken into account. For example, the DVD drive for a certain notebook might not be a valid up-sell for a desktop. Thus, it was important to create a solution capable of differentiating and evaluating cross-sells and up-sells right from the data as presented in Table 1, and of taking into account all compatibility constraints.

The nature of the SMB store drove the type of analysis the team was able to use. The store is considered public, in other words, we are unfamiliar with the customer as they come in and browse or speak to a representative. Since no login exists, and therefore the system cannot link this customer to their individual historical purchase behavior, we are unable to leverage techniques that yield individual product/offer probabilities (such as via logistic regression). Furthermore, time did not permit enhancements to the site that would call for real time data collection on the customer (i.e. technographics, clickstream behavior, or pop-up survey) for the purpose of individual-level modeling and scoring. Therefore, the team chose to adopt market-basket analyses that take advantage of the data available, despite its individual-level predictive power limitations.

Figures 2 and 3 respectively illustrate cross-sells and up-sells associated with the data in Table 1 taking into account the quantities for each SKU in the order (refer to Table 1 for a description of the SKUs). This type of graphical representation of the market basket analysis is used to simplify the interpretation of results. In the context of the example order represented in Table 1, the nodes of the graph in Figure 2 represent the SKUs and the lines along with the corresponding label represent the frequency of the paired relationship. For example, the pair C7971A: 292887-001 occurred seven times in the example order. Note that the arc's thickness is proportional to its associated frequency; hence the important relationships are represented by the thicker lines. One simple characteristic of our graphical representation is that it represents quantities using the color of the origin node, this helps distinguish the quantities associated to each arc.

Figure 2 - Cross-sell market basket analysis for order 849608

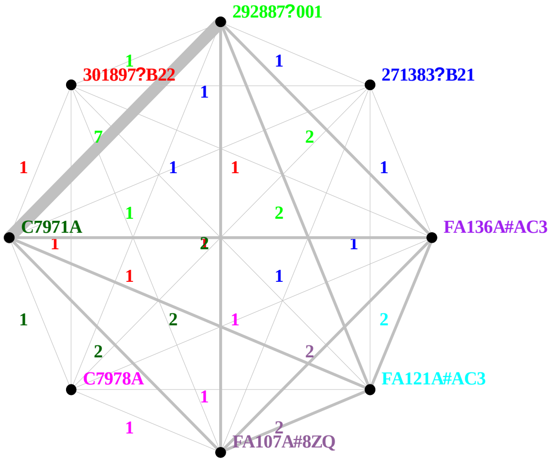
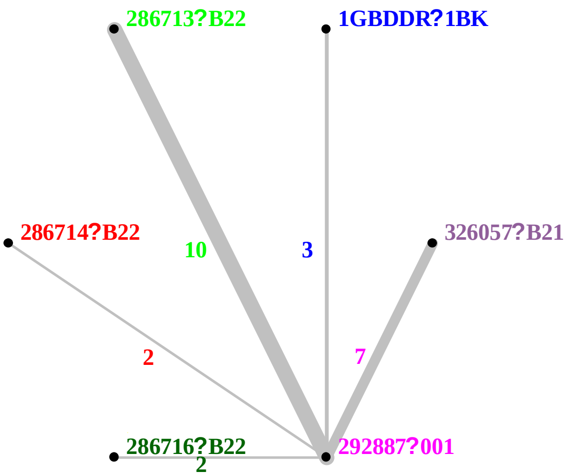


Figure 3 - Up-sell market basket analysis for order 849608



The analysis presented in Figures 2 and 3 was made for all orders of interest and results were aggregated representing the affinities found in the time span of interest. In summary, for any particular product and its associated SKUs, the solution reads from the database for all the data matching the set of selected SKUs, analyzes for up-sell and cross-sell, ranks recommendations by creating a ranking function that takes into account margin and revenue, and provides up to three alternate recommendations. Because of user navigation and page size restrictions, the business limited the number of recommendations per product to seven for up-sells and five for cross-sells.

Overall, the analysts first generated up to seven up-sell recommendations (unique to the target product) and likewise, up to five unique cross-sell recommendations. Then, an additional two alternate recommendations for each of the above generated recommendations were created to give business owners the option to overrule the originally suggested offers.

Recommendation Generation

To accomplish the formerly mentioned requirements, one function with several parameters was created to generate all recommendations for all products in the study. For example, one particular function takes product info and time-frame as an input and creates a three-tabbed spreadsheet-level recommendation file for the selected product. The three tabs are provided for the eventual need of substituting any of the winning recommendations (due to reasons such as new information regarding product availability, conflict with other promotions, or marketing's desire to feature a new product). Tab 1 includes all possible recommendations, the second tab includes only those that have a probability of acceptance above a certain acceptable limit, and the third one only includes the final recommendations.

Table 2 represents an example of the output generated by the program for a particular family of desktops. Note that the function created is completely automated in the sense that it generates an optimized list of all up-sell and cross-sell recommendations for the product in question. The program was optimized such that it usually takes less than 5 minutes to read all the data from Database, generate all recommendations for a product, and create the corresponding spreadsheet recommendation file.

Table 2: Example of the output generated (some sample fields only)

Cross-sell/Upsell	Recommendation type	Category	Offer description	Offer sku	Probability	Price	Revenue Rank	Margin Rank	Rank	Lead time
up sell	Primary	Memory	512MB PC2-3200 (DDR2-400)	PM848AV	0.51	$200	111	111	111	7
up sell	Secondary	Memory	1GB PC2-3200 DDR-2 400 (2X512)	PM842AV	0.25	$310	109	110	109.5	7
up sell	Secondary	Memory	2GB PC2-3200 DDR2-400 (4x512MB)	PM846AV	0.05	$560	100	104	102	7
up sell	Primary	Processor	Intel Pentium 4 520 w/HT (2.80GHz, 1M	PM675AV	0.36	$228	110	108	109	7
up sell	Secondary	Processor	Intel Pentium 4 540 w/HT (3.20GHz, 1M	PM677AV	0.22	$298	108	106	107	7
up sell	Secondary	Processor	Intel Pentium 4 550 w/HT (3.40GHz, 1M	PM678AV	0.16	$373	107	103	105	7
cross sell	Primary	Mobility Thin & Light	HP Compaq Business Notebook nx611(	PT602AA#A	0.04	$999	103	95	99	21
cross sell	Secondary	Mobility Thin & Light	Configurable- HP Compaq Business Not	PD875AV	0.04	$510	87	77	82	15

The optimization method takes into account several other parameters provided by the users like minimum selection probability which puts a restriction on recommendations such that all recommendations are warranted to have a minimum empirical probability of acceptance in the training data set independent of the revenue or margin associated to it. Once these low- probability recommendations are eliminated, the margin and the revenue are used to rank the remaining candidates. The ranking function implemented for this purpose is discussed next.

Note that there are three ranking columns in Table 2; *Revenue Rank* gives the position of the recommendation if these were sorted by ascending revenue such that the largest expected revenue would get the largest rank; likewise for the *Margin Rank*. The third *Rank* is the weighted average of the former two. The weights for revenue and margin were provided by business owners associated with the corresponding product category (printers, desktops etc). In mathematical form, the function implemented to rank each recommendation (R_R) is as follows:

$$R_R = W_M \times R_M + W_R \times R_R \mid (W_M + W_R = 1)$$

where,

- $W_M \in [0,1]$ is the weight given to margin
- R_M is the ranking position of recommendation R when sorted ascending by expected margin (EM_R)
- $W_R \in [0,1]$ is the weight given to revenue
- R_R is the ranking position of recommendation R when sorted ascending by expected revenue (ER_R)

In the former definitions $ER_R = R_R \times p_R$, and $EM_R = M_R \times p_R$ where p_R is the probability of acceptance associated to recommendation R. In Table 1 p_R is labeled *Probability* and is calculated using the following formula: $p_R = n_R/n_{sku}$ where n_{SKU} is the number of product SKUs sold in the time frame analyzed, and n_R = number of times that the recommendation R was selected in those n_{SKU} products sold.

The reason for selecting this relatively simple ranking formula was to provide business owners with the option of giving relative importance to revenue or margin depending on their organizational goals or needs. Some managers decided to use a 50%/50% weight split, while others put more emphasis on margin (75%). The team considered several Ranking functions, but this was selected because it was easy to get business buy-in and allowed their participation in the development of the solutions implemented for their respective business sectors. Besides, note that the header *Recommendation Type* refers to whether the recommendation for the *Category* (e.g. memory) is the first choice (Primary) as selected by the *Rank*, or an alternate recommendation (Secondary) for the same category. Thus, in general the team tried to involve product managers as much as possible in the process of selecting the final recommendations for their products.

4. Recommendation Review Process & Final File Generation

Analysts presented and reviewed the recommendations with business owners and members of North America eBusiness and segment marketing organizations to receive approval and finalize the target offer(s) to be deployed into production web-site and call-centers. The process also required agreement on the proper website placement (left hand, inline, cart) and text to be used for the offer/callout within the website.

The deliverable of the review process was up to 7 up-sell and 5 cross-sell recommendations. The recommendations were then handed over to the marketing team, which finalized the offers' wording and their respective placements.

Implementation through Content Rendering Tool & Test Design

Once the recommendation file was finalized, it was again reviewed by analytics and segment marketing teams for accuracy and then scheduled to be implemented on the SMB website at midnight of the agreed-upon date. Figure 1, displayed earlier, illustrates an example of up-sell recommendation in the configurator page and a cross-sell recommendation in the cart page.

Mentioned above, the team was restricted in time and further limited in enhancements that could be made to the site for the purpose of testing. Thus, randomization of customers upon entry to the site was not possible and the team could not have customers sent to test or control cells. Both the analyst and business teams involved understood this limitation and the groups proposed to compare pilot performance to some pre-pilot period.

5. Periodic Reporting and Final Pilot Performance Evaluation

The analytics team proposed a process to evaluate the effect of the recommendations. Before the pilot was launched, analytics team members and business owners agreed upon several metrics. Upon recommendation implementation in the web site and call center, the team monitored and evaluated them using several techniques in parallel. A final report was delivered to management regarding the weekly performance of the pilot.

Financial and Attach Metrics

Performance of each metric was reviewed at the sku level and then rolled up to the product category level (e.g. all notebooks were aggregated to a "notebook" category) and ultimately to the pilot level. Due to the aforementioned challenge regarding the lack of a control cell, a month prior to the launch of the pilot was chosen as the Control period. Some of the metrics and their performance are discussed below:

- **Attach Rate** is the ratio of the Attach orders (orders with either a cross-sell or up-sell or both) to the total number of orders for that sku.
- **Attach revenue by Sku revenue** is the ratio of Attach revenue (sum of the up-sell revenue and cross-sell revenue) to the sku revenue. It reflects for every dollar of sku how many cents of attach revenue is generated.
- **Average Order value** is the ratio of the total order revenue to the total number of orders.

Table 3 below shows the recommendations significantly impacted the metrics discussed above for most of the categories. For example, we saw an 18% increase in *Attach Rate*, a 36% increase in *Attach Revenue by SKU revenue*, and more than a 2% increase in *Average order Value* for Desktops for Phase I. Similarly, we observed significant positive outcomes in Phase II.

Statistical Significance

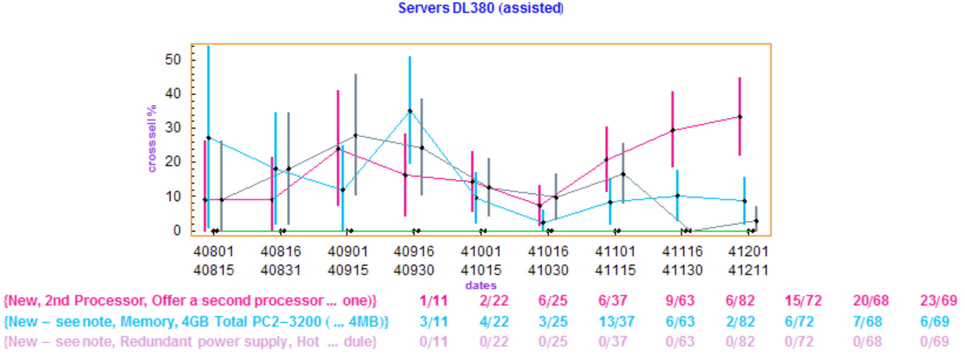
Control charts were created to demonstrate the significance of the change (if any) in proportion of acceptance for each recommended sku. Figure 4 shows an example of a graphical analysis made to evaluate the effectiveness of the recommendations. In particular this figure represents cross-sell analysis for Servers DL380 when orders were assisted by sales representatives. In this plot the x axis represents dates (40801=>2004/08/01) and each series describes the cross-sell % over time. Note that each point has a 90% confidence interval around the corresponding proportion.

These confidence intervals help evaluate whether there is a significant change in the proportion from one period to the other. For example, note that for the 2nd processor, the cross sell proportion from 10/16 to 10/30 is significantly lower than that from 11/16 to 11/30. For each product category, similar evaluations were made for combinations of: a) up-sell, cross-sell, and b) assisted, unassisted, or all sales together. All evaluations were programmed such that whenever the program was run these plots would be automatically updated for all products in the pilot. Thus, not only was the recommendation process automated, but also the process of evaluating the effectiveness for all recommendations.

Table 3 – Sample of Product Performance on Key Metrics (% change from control period)

Product Category	Attach Rate		Attach Revenue by SKU		Average Order Value	
	Phase I	Phase II	Phase I	Phase II	Phase I	Phase II
Desktops	18.2%	8.2%	36.8%	10.5%	2.2%	16.2%
Printers	1.8%	20.6%	59.3%	30.2%	26.0%	7.1%

Figure 4 – Sequential confidence intervals for testing the effectiveness of recommendations



Incremental Revenue

The team also evaluated the incremental revenue associated with each recommendation for each SKU. If there was a positive, significant change in the percentage of acceptance for a particular recommendation, we proceeded to estimate the incremental revenue associated with each recommendation. The method was as follows: Assuming that for any product SKU (SKUp) there are n recommendations SKUs ($SKUp_R$, R=1, 2…, n), the incremental revenue for each of the $SKUp_R$ is calculated using the following formula:

$$IRp_R = CRp_R \times (1 - CPp_R / PPp_R),$$

where,

- CRp_R = revenue associated to SKU_R when it is sold along with product p in the control period,
- CP_R = proportion of acceptance for SKU_R when it is sold along with product p in the control period, and
- PPp_R = proportion of acceptance for SKU_R when it is sold along with product p in the pilot period.

Note that this method is not dependent on the length of the pilot and control periods. This is true because the differences in sample sizes associated with each period are being considered statistically on the test of hypotheses to compare the proportion of acceptances in the pilot and control periods.

To illustrate the idea consider this example that does not necessarily represent real data:

−Configuration sku: DR547AV-DX2
−Offer sku: DR689AV-512MB DDR 333MHz
−Control period data: DR689AV sold in 18 out of 99 orders of DR547AV-DX2 for a 18.2%
− Pilot period data: DR689AV sold in 58 out of 161 orders of DR547AV-DX2 for a 36.0%
−Revenue associated to the DR689AV in pilot period is $53,200
Incremental Revenue associated to DR689AV = 53,200 × (1- 18.2/36) = $26,350

A program to calculate all values of incremental revenue was created such that all of these values would be automatically updated for all products when it was time to evaluate the economical impact of the project.

6. Closing Comments

Overall, the pilot generated a ROI of $300K per month, a 3% increase in attach rate, 15% lift in Attach Revenue to SKU Revenue, and greater than 5% improvement in average order value. However, the team found benefits that may be even more important than the financial ones. The relatively new team was able forge strong relationships with the business owners, educate them on the benefits of analytics, and gain their support for future data analytics ventures.

Publishing Operational Models of Data Mining Case Studies

Timm Euler
University of Dortmund, Germany
Computer Science VIII
euler[ατ]ls8.cs.uni-dortmund.de

Abstract

This paper presents a method to publish executable models of data mining case studies in a so-called Case Base, where they can be inspected in detail by anyone using a common web browser. The Case Base serves to inspire and educate other developers of data mining applications, in particular if they are not yet experts in the field. A case study can be directly downloaded and executed on new data if it is found to provide a useful template. The approach is validated and exemplified using two data mining case studies from an Italian and a Polish telecommunications company. These case studies are interesting in their own right in that they involve complex data preparation issues. The paper studies these issues and relates them to the knowledge transfer opportunities that the Case Base offers.

1. Introduction

Software tools supporting common data mining or knowledge discovery tasks have matured greatly in the past decade, offering now basic graphical and conceptual support for all phases of the knowledge discovery process. This eases the daily work of data mining experts and allows a growing number of non-experts to try and start knowledge discovery projects. Though both experts and inexperienced users may find guidelines for their work in the CRISP-DM model [5], they are still faced with two essential problems, those of finding a suitable data representation and of choosing and tuning a learning algorithm to give acceptable results. Data preparation, as the subprocess that leads to the desired data representation, still consumes the largest part of the overall work, according to [11] and a 2003 KDnuggets poll, despite the existence of graphical, data flow oriented user interfaces for this task in modern software tools. The likely reason is that what is a good data representation depends on the mining task and data at hand, which poses a challenging problem, especially for inexperienced users. Such users would benefit greatly from sources of knowledge about how experts have solved past KDD (Knowledge Discovery in Databases) problems, especially from exemplary, executable KDD solutions. Even the experts might find inspirations in solutions from other business domains if these were available to them. The need for an environment to exchange and reuse KDD processes has long been recognised in the KDD community, see section 3.

This paper uses a framework in which successful KDD processes can be modelled, executed, and published to different KDD users, to present two data mining case studies. A web platform (the Case Base) to publicly display the models in a structured way, together with descriptions about their business domains, goals, methods and results, is described. The models are downloadable from the web platform and can be imported into the system which executes them (on a relational database).

The two case studies have been published in the Case Base, and are presented in detail in this paper. The descriptions here can be compared to the models in the Case Base. Issues that were identified as relevant by the authors of the two studies when evaluating the framework are discussed. In particular, the first study allowed a direct comparison of the effectiveness of graphical process modelling as compared to manual, low-level programming. Further, the second study indicated that some advantages of graphical modelling, as in the presented framework, can be disadvantageous when used by non-experts in a naïve way. The second study also allowed to compare the processing performance of a relational database with that of a SAS installation.

The paper is organised as follows: section 2 describes the MiningMart framework which provides the metamodel in which the models of the case studies are expressed. Section 3 gives related work. Sections 4 and 5 present the two case studies, each with some discussion of relevant problems and lessons learned. Finally, section 6 concludes the paper.

2. The MiningMart framework

MiningMart is an environment for the development of KDD applications that makes use of a formal metamodel (called M4) to model KDD processes. The central ideas of MiningMart have been published in [10]. They are summarised in this section and extended by a more detailed discussion of the Case Base and its technology, since this concerns publishing the case studies which are presented in later sections of this paper.

2.1. Overview

In MiningMart, both the data and the processing/mining operations on the data are modelled declaratively using M4, and translated to operational SQL by a compiler module. A data model and a process model together describe an instance of a KDD process and are called a *case*.

The metamodel M4 can be expressed in various ways, for example a relational database schema or an XML DTD. MiningMart currently uses a database to store the case models while working with them, but uses XML for import and export of the models. A database has the advantage that good support for the consistency of the case models is given, as the dependencies between model objects can be expressed in database constraints such as foreign key links.

In order to facilitate the reuse of cases, a data model in M4 consists of two levels. On the higher level, data is modeled by *concepts*, which contain *features*, and *relationships*. Every step of the KDD process is described, in terms of input and output, on this level. The lower level uses *tables* and *columns* to which the higher-level elements are mapped. It can model both database schemas and flat file data. This two-level approach allows to reuse the higher level elements on new data by simply changing the mapping. For the mapping, each concept corresponds to one table or view, a feature can correspond to one or more columns, and relationships correspond to foreign key links between tables.

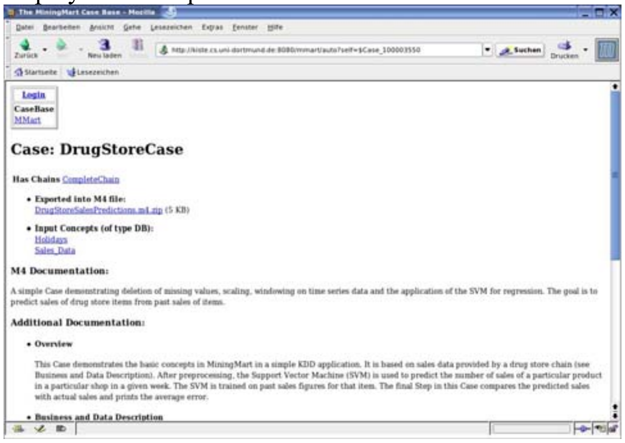
The mapping between the levels is provided by the user, if the case is developed for the first time; in the MiningMart system, a graphical editor supports the creation and manipulation of higher level elements and their mapping to given data. However, if an existing case is reused, a simple schema-matching algorithm can be employed to find at least a partial mapping. The matcher algorithm is based on comparing the names and datatypes of the concepts and features (higher level) of the existing case, and the tables and columns (lower level) of the given data (compare [12]). Once the mapping is done, all user work on the KDD process continues using the higher data level. This provides a more abstract, task-oriented view of the KDD process than low-level programming would.

To model the data processing operations, the metamodel allows to define characteristics of basic processing operations by specifying *operators*. The definition of an operator in the metamodel includes its name, its input and output parameters and their types (concept, feature, or simple values like strings or numbers), and constraints on the parameters that must be fulfilled. A typical constraint might specify, for example, that a certain input feature must have a certain conceptual datatype. The actual processing behaviour of the operator is not specified in the metamodel but in the system that interprets it. This is the *compiler* functionality of the system. The output of an operator can be used as the input to another operator, so that the data flow induces a directed acyclic graph of operators.

To ensure that a wide range of KDD processes can be modeled, new operators can easily be added declaratively to M4 and will then automatically be available in the system; only the compiler has to be extended by a new module for each new operator (a Java API is available for this task).

2.2. The case base

This section describes the knowledge portal, called Case Base (`http://mmart.cs.uni-dortmund.de`), that serves to distribute successful KDD models (cases) publicly. The core of this portal is a software called InfoLayer [8] that translates structured information, according to a given ontology, to HTML files. It can also generate RDF files which can be read by software agents. In MiningMart, the ontology is the metamodel M4, and a collection of instances of this ontology forms the central repository of KDD cases. Only the higher level of the data model is published for confidentiality reasons. These higher-level parts are represented in UML, which is read by the InfoLayer software. The UML classes are linked to a database that contains the M4 schema. Whenever a web client requests information about an M4 object (via HTTP), the InfoLayer creates an HTML file for it, disregarding caching for this discussion (M4 objects are *operators*, *concepts* etc.). The HTML files are generated using templates that provide the layout for the information to be displayed. There can be zero or one layout template for each type of M4 object. If no template is given, the contents of an HTML file for an M4 object are automatically determined by the InfoLayer software from the UML model. A template can be used to provide only parts of the default contents, or to arrange them in a particular way, for example by using HTML tables. By default, the M4 object is displayed with its name, its properties, and the names of M4 objects it is directly linked to. The linked M4 objects appear as HTML links so that a web user can browse through a case model according to the structure of M4. For instance, an operator is displayed together with its name and its parameters, and a click on any parameter shows the realisation of that parameter, which is in turn an M4 object, for example a concept used as an input parameter for the operator. The following is a screenshot showing the case base as it displays an example case.



Case: DrugStoreCase

Has Chains CompleteChain

- Exported into M4 file:
  DrugStoreSalesPredictions.m4.zip (5 KB)
- Input Concepts (of type DB):
  Holidays
  Sales_Data

M4 Documentation:

A simple Case demonstrating deletion of missing values, scaling, windowing on time series data and the application of the SVM for regression. The goal is to predict sales of drug store items from past sales of items.

Additional Documentation:

- Overview

  This Case demonstrates the basic concepts in MiningMart in a simple KDD application. It is based on sales data provided by a drug store chain (see Business and Data Description). After preprocessing, the Support Vector Machine (SVM) is used to predict the number of sales of a particular product in a particular shop in a given week. The SVM is trained on past sales figures for that item. The final Step in this Case compares the predicted sales with actual sales and prints the average error.

- Business and Data Description

When setting up a case with the MiningMart system, every object from the case itself to operators, parameters, concepts and features can be documented using free text. These comments serve users for their own orientation in complex models. They are stored in M4 and appear on the web pages when a case is published, so that other users browsing the case have a good orientation as to the purpose of each step in the KDD model and the use of their parameters. If such comments are missing, they can be added by the operators of the case base.

However, users who search for a case which they might use as an inspiration for their own KDD problem, or even as a blueprint of a solution, need some additional, more general information about each case. The most important types of information are (i) the business domain, (ii) the business problem that was attempted to solve, (iii) the kind of data that was used, (iv) the mining task and other KDD methods that were employed, and (v) the results, both in terms of KDD and the original business problem. Hence, exactly this information is provided together with every case that is presented in the case base. To this end there is a template with five slots for free text, corresponding to the five types of information above, which is to be filled by every case publisher (a sixth slot with contact information enables further inquiries by interested users). The filled template is displayed in the case base as the first page of information about each case. From there users who are motivated by the descriptions can start to browse the case model, beginning with the chains of operators or the input data. In this way, the case model is related to the context in which it was set up, which allows to judge its suitability for a new business problem. Finally, each case model is linked to a file that can be imported into a MiningMart client.

2.3. Case retrieval

This section briefly discusses a few ideas for case retrieval, that is, how to find a MiningMart case that can serve as a template for an own solution from the case base. A suitable starting point is the additional documentation published in the case base for every case. Assuming a low number of published cases, this information can be searched manually, but as the case base grows, automatic search methods should be added to allow at least keyword search. Another useful way of approaching the case base can be offered by sorting the cases according to various topics extracted from the additional case documentation. The five slots of the documentation template provide five useful topics for indexing the case base. Further topics (such as type of business/institution where the application was realised) can be added by extracting this information from the free text descriptions in the slot.

The business-related information will often not be enough to determine whether a published solution is suitable for adaptation to own data sets. A second method of approaching the case base is by looking for data models in it, called *target models* hereafter, that are similar to the own (local) data sets. The automatic schema matcher included in MiningMart can be used for this. It searches among all data models in the case base for models similar to the local data.

This online method has an important advantage. All cases use a particular data model as input, then preparation operations are applied to the data. Each preparation operation produces intermediate data models. These intermediate models can be included into the search for target models, so that the most suitable *entry point* into a case can be found. Since preparation is actually a method to adapt data representations, it would make no sense to restrict the search for target data models to the initial data that the original KDD process started out on. Schema matching is a useful tool in this setting as the number of target data models is high, making manual search for the best entry point a cumbersome task.

A unique option that the case base offers is to search it for common subtasks that have been solved using identical processing structures. A simple subgraph detection algorithm can be used for this (since the nodes of the graphs are typed, efficient algorithms exist). More cases are needed before this will lead to interesting results, however.

3. Related work

MiningMart was mainly described in [10]; see also related work cited there. The technology of the case base was updated recently; this and case retrieval issues are a contribution of this paper. The idea of collecting and publishing KDD solutions was mentioned (though not realised) early in [15] and [9]. The importance of the reusability of KDD models is also stressed in [16] and [2].

To document and store KDD processes requires a modeling language, or metamodel. A well-known but informal standard to model the KDD process is Crisp-Dm [5]. The new PMML version 3.0, a standard to describe machine-learned models in XML [13], includes facilities to model the data set and data transformations executed on it before mining. However, it is not process-oriented, thus it does not allow to model a data flow through a complex KDD process, and the data model is restricted to one table. Other standards around data mining are Java Data Mining and SQL/MM Data Mining. Though extensible, they currently provide interfaces to modeling algorithms rather than to complete KDD processes. Similarly, in [3] a data mining ontology is presented to enable grid-based services, but is currently restricted to the modeling phase of the KDD process.

Recently, some new research attempts to employ grid infrastructures for knowledge discovery; a good overview is given in [4]. To enable the execution of KDD processes on a grid, these processes have to be modeled independently from the machines that execute them, and heterogenous data schemas and sources have to be modeled. In [1], a Discovery Process Markup Language (DPML) is used, based on XML, to model the complete KDD process. Unfortunately, from the available publications it is not clear how comprehensive and detailed DPML is.

4. Case study 1: Churn prediction

This section describes a data mining application that was developed in an Italian telecommunications institute. An overview of it was given in [7] and [14]; the present paper adds important details as regards the data preparation and the lessons learned.

A major concern in customer relationship management in telecommunications companies is the ease with which customers can move to a competitor, a process called "churning". Churning is a costly process for the company, as it is much cheaper to retain a customer than to acquire a new one [14]. Churn prediction is the task of predicting which types of customers are likely to churn, and more challenging, when they will churn. These business problems can be translated to data mining or KDD problems in various ways. One successful translation to a classification task that predicts a class of customers likely to churn within a given month in the near future is described in this paper. The task was solved using decision trees which achieved a predictive accuracy of 82%. This good result was only possible due to the introduction of relevant derived features for prediction which were not available in the original data, and due to a re-representation of the data so that temporal aspects could be included. Thus data preprocessing was a key success factor in this application.

One interesting aspect of this case study is that it was implemented twice, based on manual programming on the one hand, and on graphical modelling on the other. This allowed to compare the amounts of work spent by highly paid KDD experts on the application in both scenarios (see section 4.6).

4.1. Overview

As said above, the objectives of the application to be presented here were to find out which types of customers of a telecommunications company are likely to churn, and when. To this end, the available data tables were transformed so that a classification algorithm could be applied. In the resulting data set, each row (that is, each example for classification) corresponded to one customer of the company, and contained many features describing their telecommunication behaviour for each of five consecutive months. Whether or not the customer left the company in the sixth month determined the classification label or target. Thus a binary classification problem was formed that could directly be addressed using several classification algorithms. Once a learned classifier is available it can be applied every month on data from the current and past four months, to predict churn for the following month. A longer prediction horizon (to predict churn not for the following month but, say, the second or third month) can be realised easily by changing a few parameters in the graphical model of the application.

4.2. The data

The available data sets were: (i) call detail records, recording for each phone call a customer made the time and date, called number, tariff, type of call etc.; (ii) billing data from the accounts department, containing revenues generated by each customer in a given month; (iii) and customer services data from the customer registry, containing the gender and address of a customer as well as the dates of entering and leaving their contract with the company. Those customers still with the company serve as negative examples for churn, while for those who have left the company, the data from the last five months they stayed with the company is used to form positive examples.

4.3. Data preparation

The first table to be prepared is the call detail records table. The transformation of the original data starts by extracting an Id for each month from the date of each call, because monthwise statistics are needed. This month Id has to be the same as the one used in the billing data. A new column with the month Id is added to the call detail records. Additionally, the type of phonecall (internet provider, local, distance, abroad, mobile phone etc.) is derived from the number called, with the aim of creating a telecommunication profile for each customer.

Next, the time span to be used for mining (the five consecutive months) must be selected from the complete table. Those customers who happened to have left the company at the end of the time span are positive examples for churning. However, selecting only one particular time span does not deliver a sufficient number of positive examples. Also it might introduce a learning bias. For these reasons, six different spans were selected. Notice that the six resulting data sets are likely to contain overlapping customer sets, since many customers (who have not left the company) participate in all time spans.

Now the six subsets must be mapped to the same time index (e.g. 1 to 5 for the five months), so that the six time spans can be matched. After creating the common time index, the six data sets are further processed in exactly the same way. Rather than setting up the same transformation process six times, one might set it up once and use it on six different inputs. However, the overall mining process should be executable automatically every month to predict new groups of churners. Not every KDD system supports automatic execution of a modelled process on different input tables. In this application, a different approach was taken that simplifies the automatic execution by exploiting a *Segmentation* operator available in MiningMart. One additional, nominal column is added to each of the six data sets that contains only one value which is different for each data set. Then the data sets are unified (using a union operation like in SQL). Now the segmentation operator takes the additional column as the segmentation index, and ensures that all following operators are applied to each segment in parallel. This means that the process can be described hereafter as if it was applied to only one input table, though there are six segments to be processed. This input table now contains one row per phonecall made, and four columns: the customer Id; the month Id; the calllength; and the type of call. Using aggregation as in SQL (forming a data cube), the sum of calllengths spent by every customer in each month for each type of phonecall can be computed. The resulting table contains the data that is needed for mining; however, the format is not suitable yet: it is not a table with a single row for every customer, but with 35 rows per customer: the number of months, five, times the number of different call types, seven. What is needed now are 35 new attributes: one per month and per call type. Each new attribute will contain the calllengths spent by every customer in that month making that type of phonecall.

These attributes can be created using 35 derivation operations. However, exploiting the special MiningMart operators *Segmentation* and *Pivotisation*, the process becomes much simpler. Segmentation is applied a second time, this time using the call types as the segmenting attribute. Now again the further process can be described and set up as if there was only one input table, although in reality there are 42 tables with the same data format: seven, the number of call types, times six, the number of time spans.

At this point the operator *pivotisation* can be used to gain the final representation of this part of the data in one single step. Conceptually, pivotisation creates a table with six columns, one per month plus one customer Id, so that each row corresponds to exactly one customer. Behind this conceptual view are 42 data tables, six time spans for each of the seven call types. In the next step, the union of all data tables corresponding to the same call type can be formed, leaving seven tables. Finally, the seven tables (each with five non-key attributes) are joined, resulting in one real data table with the customer Id column and 35 telecommunication profile columns, where each row contains all of the information for exactly one customer.

All of the above concerned only one of the three original data tables, the call detail records table. The second table contains the revenues per month and per customer. This table is transformed in a similar way, using selection of the six time spans and one pivotisation so that the resulting table contains one column per each of the five months, indicating the revenue generated by every customer during that month. The third table with the individual customer information contributes the target attribute: those customers who left the company in one of the six end months of the six time spans are positive examples for churning, all others are negative examples. All three preprocessing results can then be joined to form a final table with 41 columns to be used for prediction, plus the target and the key column.

4.4. Data mining

By following the rather complex preparation process above, it was possible to transform the data from a “transactional” format, containing information for every single customer transaction (phonecall), to an aggregated format from which the time-related information was still available, but which provided each customer as a single learning example.

On this table a decision tree was trained to predict the binary target. However, the first results were not satisfactory. A possible reason was presumed to be the fact that the five months that form a time span were not related to each other from the view of the learning algorithm. It was felt that *changes* in telecommunication behaviour might be a good indicator for churning, but that these changes could obviously not be found by the decision tree. Therefore additional columns were derived. As an indicator of change, the differences in the calllengths between consecutive months were tested as well as the slope of a line connecting the first and fifth month's calllengths on a graph (in this case, the sum of calllengths of all call types). This latter indicator in particular helped to increase the predictive accuracy to a satisfactory level. Note that it was only possible to use this indicator based on the complex data preprocessing phase described above.

Another factor that increased the predictive accuracy was the differentiation of customer groups according to the overall revenue that the company generated from them. More precisely, four different decision trees were trained on four groups of customers ranging over low, medium, high and very high profitability, where profitability was indicated by the sum of revenues in the five months considered. This turned out to be a successful differentiation, in that the average predictive accuracy of the four trees was 2% higher than that of one global tree trained on all customers.

4.5. The published case study

The reader is invited to compare the above descriptions to the browsable model of the case study, which is available in the MiningMart Case Base (URL see section 2.2) under “Model Case Telecom”.

4.6. Lessons learned

An interesting aspect of the case study above is that the application was implemented twice, once manually in SQL and once using MiningMart. Thus it was possible to quantify the amount of work saved by using a high-level modelling software with a GUI, compared to low-level programming. While programming the application required 12 person days, modelling it graphically could be achieved in 2 person days of work. Especially the availability of rather advanced preprocessing opertors for segmentation and pivotisation eased the task greatly in the graphical approach. Further advantages of graphical modelling that were attested are simplified documentation, especially for in-house education, simplified changing and testing of parameters (for example to change the prediction horizon, by no means a trivial change given the complex preprocessing phase), and versioning of the developing process model.

It was also confirmed in experiments that the overhead caused by parsing and translating the declarative model is negligible for real-world data sets (in this application, two million records were processed). Since MiningMart translates the process model to SQL, this approach scales as far as the underlying database scales. However, an interesting counterpart to this situation is encountered in the second case study, compare section 5.3.

This data mining application led the company that commissioned it to execute a trial campaign on a sample of the customers, to reduce churn. The results justified the investment in the project. The application was therefore integrated into other measures for CRM in the front-back office automation of that company.

5. Case study 2: Targeting a marketing campaign

This section describes a marketing application in the Polish telecommunications institute NIT. A technical report about it is available [6].

5.1. Overview

The task that was solved in the application was customer profiling, for the purpose of adapting a marketing strategy to introduce a new voice mail product. Three sources of data were available: call detail records, listing information about each phone call each customer has made; contract data, listing information about the type of contract each customer has signed with the company; and data from a call center that contacted a number of customers initially to see if they would buy the new product.

From this data, after a long process involving dozens of atomic data transformations, the input to the mining algorithm was constructed. Here only the key points of the process are described.

The biggest part of the data preparation was consumed by processing the call detail records, rather like in the first case study (section 4). This table contains the start time, date, length in minutes, number of tariff units, the number called and some other information about each phonecall of each customer. This large amount of detailed data had to be aggregated to meaningful single statistics for each customer. This was done in a similar fashion as in the other case study. However, the first attempt to do so involved segmenting the data such that each customer corresponded to a single segment. Conceptually, in the graphical model that MiningMart provides, this is a neat way of setting up a short processing chain to solve the given problem. Technically, however, this means to create as many SQL views on the original table as there are customers stored in it. This approach does not scale so well to larger amounts of data. In particular, the overhead caused by compiling the declarative process model into SQL, which was found to be negligible in the first case study (section 4.6), was very high in this study under this approach, due to the high number of views that had to be created and, in progress, evaluated. Changing the conceptual setting such that first some aggregation tasks were performed, and only then the segmentation took place, was therefore beneficial.

The customer profiles built in this way were used to predict the customers’ response to the new product, based on the call center data. This data provided the target attribute for prediction: whether the customers responded positively or negatively to the product offer. Since only a small sample of customers could be contacted by the call center, mining was used to generalise from the sample to the whole group of customers, in order to save marketing costs. Detailed results are unfortunately kept confidential.

5.2. The published case study

The reader is invited to compare the above descriptions to the browsable model of the case study, which is available in the MiningMart Case Base (URL see section 2.2) under “Call Center Case – NIT”.

5.3. Lessons learned

Two interesting aspects of this case study were identified. The first one is described above, and concerns the scalability problem encountered using the naïve segmentation approach. It shows that although many tools provide rather high-quality, high-level support for data processing, still one needs experts who know the underlying procedures well enough to develop efficient models. This emphasises the need for a public repository of successful KDD solutions, such as the Case Base presented in section 2, to provide templates or blueprints that help new developers of KDD applications to avoid traps that others have already discovered. As powerful KDD tools are becoming increasingly available and easy to use, knowledge about good KDD solutions must not stay behind, but needs an efficient means of distribution such as the public Case Base.

The second interesting aspect of this study is that it was implemented also both in MiningMart and in another system, namely SAS. Since MiningMart translates its models to SQL, this allowed to compare the data processing performance of the underlying relational database (Oracle) with that of SAS. Sound claims about the relative performances of these two environments cannot be made because they were installed on different hardware. However, data processing was about two times faster in the SAS system, which is not surprising given that relational database management systems must perform overhead tasks such as consistency and rollback management. On the other hand, SAS required manual programming of many tasks that could be realised easily in the GUI using MiningMart. Further, having intermediate processing results available in the database allows simpler examination of such results by querying and searching. These contrasting issues have to be prioritised based on the intended application when choosing a suitable data mining workbench.

6. Conclusions

This paper has presented two case studies in data mining from the area of telecommunications. The focus was on data preparation, where usually the bulk of efforts in a mining project is spent. In both studies it was possible to reduce these efforts greatly using a powerful, graphical preprocessing environment. However, the second study showed that such environments do not render experts in the field superfluous, but that detailed knowledge of underlying processes is necessary to develop a successful application.

This, in turn, motivates the introduction of a declarative metamodel for modelling such successful applications; using the metamodel, the (annotated) application models can be published to be inspected in detail by anyone. This serves to distribute knowledge about successful case studies to less experienced users, and helps them to avoid hidden traps. Of course, also unsuccessful applications can be documented in the Case Base.

The two studies also show that a choice of the most suitable environment to conduct an application in is dependent on several criteria, which may be prioritised differently in different applications. Scalability of the underlying processing software is important, but the possibility to use graphical modelling in a suitable user interface can help to save developing time (compared to low-level programming), and developing time is usually more expensive than computing time.

7. References

[1] S. AlSairafi, F. Emmanouil, M. Ghanem, N. Giannadakis, Y. Guo, D. Kalaitzopoulos, M. Osmond, A. Rowe, J. Syed, and P. Wendel, “The Design of Discovery Net: Towards Open Grid Services for Knowledge Discovery”, *High-Performance Computing Applications*, 17(3), pp. 297-315, 2003.

[2] A. Bernstein, S. Hill, and F. Provost, “Toward Intelligent Assistance for a Data Mining Process: An Ontology-Based Approach for Cost-Sensitive Classification”, *IEEE Transactions on Knowledge and Data Engineering*, 17(4), pp. 503-518, 2005.

[3] M. Cannataro and C. Comito, “A Data Mining Ontology for Grid Programming”, *1st Workshop on Semantics in Peer-to-Peer and Grid Computing at the 12th International World Wide Web Conference*, 2003.

[4] M. Cannataro, A. Congiusta, C. Mastroianni, A. Pugliese, T. Domenico, and P. Trunfio, “Grid-Based Data Mining and Knowledge Discovery”, in N. Zhong and J. Liu (eds.), *Intelligent Technologies for Information Analysis*, Springer, 2004.

[5] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, “CRISP-DM 1.0“, Technical Report, The CRISP-DM Consortium, 2000.

[6] C. Chudzian, J. Granat, and W. Tracyk, “Call Center Case”, Deliverable D17.2b, IST Project MiningMart, IST-11993, 2003.

[7] T. Euler, “Churn Prediction in Telecommunications Using MiningMart”, *Proceedings of the Workshop on Data Mining and Business (DMBiz) at the 9th European Conference on Principles and Practice in Knowledge Discovery in Databases (PKDD)*, 2005.

[8] S. Haustein and J. Pleumann, “Easing Participation in the Semantic Web”, *Proceedings of the International Workshop on the Semantic Web at WWW2002*, 2002.

[9] R. Kerber, H. Beck, T. Anand, and B. Smart, “Active Templates: Comprehensive Support for the Knowledge Discovery Process”, in R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro (eds.), *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 1998.

[10] K. Morik and M. Scholz, “The MiningMart Approach to Knowledge Discovery in Databases”, in N. Zhong and J. Liu (eds.), *Intelligent Technologies for Information Analysis*, Springer, 2004.

[11] D. Pyle, *Data Preparation for Data Mining*, Morgan Kaufmann Publishers, 1999.

[12] E.Rahm and P. Bernstein, “A Survey of Approaches to Automatic Schema Matching”, *The VLDB Journal*, 10, pp. 334-350, 2001.

[13] S.Raspl , “PMML Version 3.0 – Overview and Status”, in R. Grossman (ed.), *Proceedings of the Workshop on Data Mining Standards, Services and Platforms at the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2004.

[14] M. Richeldi and A. Perrucci, “Churn Analysis Case Study”, Deliverable D17.2, IST Project MiningMart, IST-11993, 2002.

[15] R. Wirth, C. Shearer, U. Grimmer, T. Reinartz, J. Schlösser, C. Breitner, R. Engels, and G. Lindner, “Towards Process-Oriented Tool Support for KDD“, Proceedings of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery, 1997.

[16] N. Zhong, C. Liu, and S. Ohsuga, “Dynamically Organizing KDD Processes”, *International Journal of Pattern Recognition and Artificial Intelligence*, 15(3), pp.451-473, 2001.

Championing LTV at LTC

Edmund Freeman
Washington Mutual Bank
1201 3rd Avenue
Seattle, WA 98101
(206) 461-7883
edmund.freeman[ατ]*wamu.net*

Gabor Melli
PredictionWorks Inc.
#6717 – 37th Ave SW
Seattle, WA, 98126
(206) 369-3582
gmelli[ατ]*predictionworks.com*

ABSTRACT

In this paper we report on the successful implementation of a life-time value (LTV) forecasting system at a large telecommunications company. While some research results have been reported elsewhere on the technical challenges of modeling customer value, our experience suggests that a data mining system implementation can expect to encounter several organizational challenges that can impinge on its success. We provide a background on the application, and then analyze several success factors.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

General Terms

Management, Economics, Human Factors.

Keywords

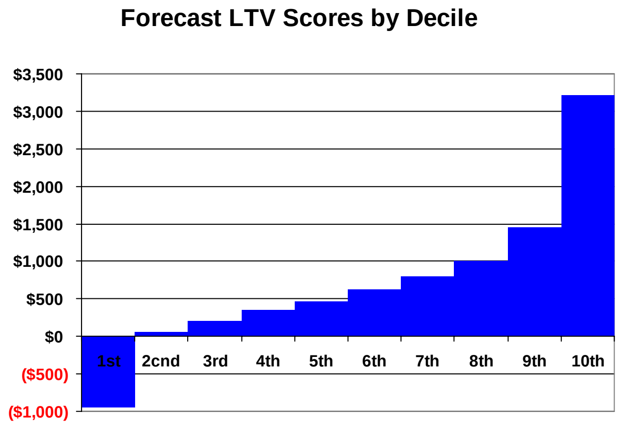
Organizational Change, Data Mining, Customer Lifetime Value, Successful Data Mining.

INTRODUCTION

One of the application areas that data mining has begun to assist in industry is in the forecasting of customer value [3,6]. The application is particularly useful in the service sector where a large variance exists between the revenue generated by a customer and the cost associated with delivery of service to that customer [4]. The differential between these two monetary figures, over a customer's entire relationship with the organization, is often referred to as the customer's life-time value (LTV, or sometimes as CLV). Figure 1 illustrates the breadth of the range in customer value at a telecommunications company. Noticed that a small but substantial portion of the customers have negative value. This situation can occur for example when a customer makes extensive use of an unprofitable feature that exists in order to match a competitor's offering. If an organization where able to identify these negative valued customers in advance then it could respond by not incurring additional optional costs on these customer. The organization could for example, avoid the cost of mailing promotions to these customers.

In 2002, we participated in just such a project at a large telecommunications company (LTC). LTC was a telecommunications company with approximately 18M customers and $12B a year in revenue. Because much of their revenue came from a subscription-based service, one of LTC's larger internal groups was its customer relationship management (CRM) division. The goal of CRM is to increase revenue and customer satisfaction by keeping customers informed about services and offers that will appeal to them. Effective communication however has an incremental cost. It would be helpful if the company could predict which to which customer it would make economic sense to contact.

Figure 7 –Distribution of customers by their future



value, divided into ten equal sized groups (deciles). The negative-value customers account for approximately 8% of the customer base.

Based on a preliminary analysis we estimated that implementing the tactic of not promoting negative-value customers would, for a given marketing campaign, decrease campaign-related expenses by 5% while increasing the campaign returns by 10%. Figure 1 illustrates this segment of the customer base. The analysis also demonstrated that it was feasible to predict life-time value with sufficient accuracy. The remaining question was the cost of implementation. The estimated system cost was for approximately 0.1% of CRM's annual budget. The system would pay for itself within two months, and generate a 500% ROI.

Easier said than done - despite the significant opportunity several critical challenges were encountered both during the implementation and soon after its launch. The project required many departments to re-think LTC's relationship with their customers and the departmental relationships within LTC. Based on past experience, we suspect that any successful LTV project will impel such an organizational re-evaluation.

The first portion of the paper presents an overview of the technical challenge that the implementation team faced. The remainder of the paper presents on the less well reported aspect of maneuvering the successful implementation of such a project through the organization.

LTV SYSTEM DESIGN

The requirements of the LTV system were to produce a set of values and scores for each customer on a monthly basis. The main outputs where the following:

Forecasted Value: The remaining monetary value predicted for a customer. For example, a forecast value $3,721 for customer *x* would be an accurate forecast if when this customer terminated their account the account would have resulted in an additional $3,721 in value.

$$Forecast = \sum_{t=1}^{t=60} M(1-v)^t(1-i)^t \big/ (1+r)^t$$

Where M is the average profitability over the past 12 months, v is the probability of voluntary attrition per month, i is the probability of involuntary attrition per month, and r is the Net Present Value rate.

Past Value: The profitability to-date, including acquisition cost, for a customer. For example, a past value of (minus) -$73 for customer *y* signals has not yet resulted in a profit. LTC had substantial acquisition costs, in the $350 - $550 range per customer; it took on average two years for a customer to become profitable. Past Value was measured by keeping a running sum of monthly profitability minus the acquisition costs, and acquisition costs were measured by acquisition channel and quarter.

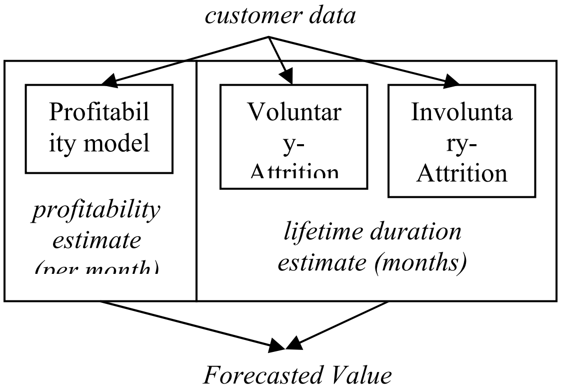
Expected Lifetime Value: The total value expected for this customer. This value is simply the summation of forecasted value and past value.

Potential Value: Another derived measure that proved to be useful was the Forecasted value with no attrition. Because Potential Value does not involve attrition, it was easy to overlay attrition data onto the Potential Value and create a cluster of high-potential, high-attrition customers to focus retention efforts on.

LTV Model

The approach taken to modeling LTV was to divide the task into three separate optimization problems. The first task was to model the customer's average expected monthly profitability. The second and third optimization tasks modeled different scenarios of when the customer would cease to be a customer. The first scenario was where the customer requests that service be terminated. This is referred to in the industry as voluntary churn. The second scenario was where the company requires that the customer's service be terminated. This is referred to as involuntary churn, and is typically the result of non-payment. Together, along with a Net Present Value factor, the three models where combined to calculate life-time value [Figure2].

Figure 8 – Data flow for the calculation of the LTV Forecasted Value.



More sophisticated approaches were considered. For the attrition models a survival model such as the Cox proportional hazards model [1] would be a relevant methodology to attempt.

The simplicity of the model did have a substantial benefit after the project was completed it was simple to modify the scores to meet special needs. For instance, 'what-if' scores where quickly introduced simply by removing off-network expenses and bad debt expenses so that CRM analysts could see the value of addressing those issues.

Definition of Profitability

We started out with the financial statements and broke expenses down into categories. We had received mixed advice about using financial statements at all, instead of having a general survey on expenses. Using hard financial statements made the LTV formulas very solid and freed us from organizational misperceptions about customer value. In particular, we found out that the critical expenses were off-network and bad debt, which very few people in the company were concerned with. The expenses people were obsessed with, namely customer care and promotions, were found to be relatively minor.

After we established the categories we decided (1) how to best allocate that category to customer activity and (2) if we should include that category at all.

Expense	Allocation	Inclusion
Off-Network Time	Minutes off network multiplied by the fee charged for the individual network used.	Yes
On-Network Time	Minutes on network multiplied by network maintenance charges.	Yes
Long-Distance	Minutes of long distance usage times a per-minute charge.	Yes
Customer Care Calls	Number of calls to care multiplied by a cost per call.	Yes
Bad Debt Expense	Revenue weighted by probability of default.	Yes
Misc. System Expense	Percent of gross revenue.	Yes.
Misc. Expenses	Flat amount per customer.	Yes
Network Capital Expense	Minutes on network multiplied by network capital expenses.	No!

The critical issue was handling the capital expense. The members of the Finance division wanted the capital expense included. We did not include this term however because the change would result in 25% of the customer base as having a negative value, instead of the more reasonable 8%. This change would have resulted in substantial operational difficulties for us.

In one sense 0 is an arbitrary number. However, when presented with a profitability analysis the natural inclination is to do the profitable things and not do the unprofitable things. Any successful system should work with this inclination and not against it. Before implementation we needed to understand the effects of identifying portions of our customer base as negative-valued.

Profitability Death Spiral

One of the lessons for future projects is that the intuitive application of data mining scores can lead to undesirable consequences. A nice example comes from a manufacturing setting [personal communication]. The company in question ran several plants at over-capacity and other plants at 60% capacity. It also possessed a relatively accurate profitability model. In the costing model however, capital expenses were allocated by unit produced. The over-capacity plants had a much lower unit cost than the plants running at 60%. What had happened was that when the profitability data was first published there were minor variations in production, and so minor variations in per-unit profitability. Naturally the company increased production in the more-profitable plants and decreased production in the less-profitable plants. The random variations were amplified, until the highly inefficient situation my colleague found was the result.

LTC was managed to attrition data, so what we did was to estimate the likely effect on attrition of removing 8% and 25% of the population from the two major campaigns, contract renewal and equipment upgrade.

Approximately 60,000 customers responded to the contract renewal program each month, and 42,000 customers a month received an equipment upgrade. Then each month we would have

	8% Negative	25% Negative
Contract Renewal	60,000	60,000
Effected	4,800	15,000
Attrition Rate	16%	16%
Extra Attrition	*768*	*2,400*
Equipment Upgrade	42,000	42,000
Effected	3,360	10,500
Attrition Rate	35%	35%
Extra Attrition	*1,176*	*3,675*
Total Extra Attrition	*1,944*	*6,075*
Attrition Increase	*1 basis points*	*4 basis points*

Neither formula would cause unmanageable problems with customer attrition. However, there were two powerful operational reasons to chose the 8% negative solution and not include capital expense in the formulas.

LTC was constantly managing to attrition and having fairly reliable attrition crises. At 25% negative it becomes a persuasive argument that we should ‘turn off’ LTV and make save offers regardless of value. At 8% negative it becomes much more reasonable to craft attrition solutions within the LTV system.

Second was the type of customers that were identified as negative. Without capital expenses each customer that was negative had a clear profitability-destroying behavior. With capital expenses a large class of negative-valued customers were those that were simply using most of their plan minutes, and our business partners were very reluctant to negatively impact those customers.

Features of the Profitability Calculation

Our profitability formula had a number of important features that were critical to the success of the project.

1) The negative-valued customers were justifiably negative-valued. For each such customer, we could identify concrete behaviors as to why they were negative-valued and that LTC would in fact be better off without that customer. As can be seen in Figure 1, the negative-valued decile is clearly negative-valued.
2) For each critical component of profitability we could give concrete advice on how to improve it. For instance, for Bad Debt expense we could give suggestions on how to acquire more credit-worthy customers.
3) The formula was based on actual financial data, so we could make meaningful comparisons between customer value and marketing offers.

PROJECT PRECURSORS

This was not the first attempt at LTC for an LTV system. Two earlier attempts did not achieve a return on investment. One was technically successful but achieved minimal impact on the business; the other did not get past the proposal stage.

Finance: Ignored Valuation

The LTC Finance department produced a customer-based valuation. They published this information by rolling all the information up to the rate plan level and then producing profitability numbers. This information was ignored outside of Finance. This was because

1) Profitability by rate plan was not the perspective taken by others in the organizations.
2) The report was painfully dense; LTC had over 1,000 active rate plans.
3) There was no way to drill down into the data and identify causes of profitability and unprofitability.
4) The calculations used base averages and were not adjusted for customer behavior such as different attrition rates and non-payment rates. For instance, Bad Debt was allocated as a percent of gross revenue. Telling people to decrease bad debt by decreasing revenue is not very actionable advice.
5) Finance made no effort to get their results out into the company and have it be used.

This first attempt proved helpful later to better understand the financial dimension to calculating life-time value.

The Consultant-Lead Committee

Another attempt at an LTV system was a consultant-lead committee (CLC). Its approach to implementing an LTV system was to interview a large number of business-oriented subject-matter experts about all aspects of customer profitability. The end result was a long report that was soon shelved. This was likely because:

1) The proposal contained hundreds of recommendations that, while grounded on the experience of subject-matter experts, needed to be pared down to a more cost-effective subset of requirements.
2) The recommendations were not grounded on a theory of customer value. Instead the metrics were based on subjective opinion and as a result was unable to defend or explain its rationale.
3) Most critically, the CLC did not have the technical expertise to implement their LTV system. This is the real reason the project never got beyond presentationware.

One source of value from this attempt at LTV was ideas on how to present LTV results.

Precursor Lessons

Both previous projects ultimately failed because they did not result in data that was useable to the enterprise. If we wanted our project to be successful we needed to make the data available, which meant we needed to get the LTV scores into the data warehouse, which meant we needed IT funding, which meant we needed to go through LTC's new funding process.

IN ORDER TO GO LIVE

The technical aspects of the project took the team approximately one third of the year that it took to launch the LTV system. The rest of the time was meeting with partners, discovering who we did not need to meet with, getting support from the people we needed, going through IT, educating the company on what LTV was and how to use it, and building ancillary systems.

The LTV project brought the project team into conflict with

- New cost control methodologies in the company,
- Finance, in terms of how marketing campaigns were evaluated,
- More Finance, in terms of how corporate profitability was managed,
- IT, in terms of how data was managed and produced,
- A Small Influential Business Unit (SIBU), and
- Regional marketing units.

Why go through this risky effort when the LTV scores could have been generated on a high-end PC? Because the project would have been unsuccessful. A successful project required that the data be housed in the company data warehouse because this made the data universally accessible and also gave the LTV project a kind of official stamp of approval.

The Start: The New Cost-Control Process

The project's initial challenge was the process for controlling operational costs, specifically the information technology budget. The process for funding new projects called for strict return on investment (ROI) calculations, with a senior manager held accountable for delivering the ROI, and the process validated by a committee selected from all units in LTC.

An ROI requirement is commonplace and reasonable but not necessarily a rational exercise. The committee members for example only had sufficient time for a cursory glance to the proposals. A consequence was that each member responded to the marching order of "get their department's projects funded", so project funding was less a matter of project merit than political connections and a friendly accountant who would give a favorable ROI evaluation.

Estimating project return presented us with a substantial difficulty. Marketing campaigns had been measured on either 1) attrition improvements or 2) new purchases. Gradually improving the behavior of the base was not in their formulas. The Finance customer valuation method allocated most costs on a per-customer basis, ignoring facts such as bad debt expense tends to be highly concentrated on customers who stop paying.

This was one of the points in the process when the use of a dedicated PC to produce LTV reporting was considered. However, the Marketing Vice-President (MVP) insisted that no such skunkworks project be done, and that we needed official funding.

In the end, the MVP figured how much ROI would be necessary to get the project through, what kind of attrition reduction would be necessary to get over that hurdle, and then promising to delivery that reduction. This was a complete prevarication; the LTV project was designed to vastly improve customer profitability at the cost of a slight increase in attrition. However, it was enough to get the project funded, at which point we ran into the Finance and IT department challenges.

Finance Department

Finance thought that an LTV project was a great idea. However, they were upset that Marketing was doing it and not them. They were already publishing a form of customer valuation, but because the information was not actionable it was not used. The situation between the two groups was understandably very tense. The situation was made unavoidable by our executive sponsor's insistence that we have Finance's formal approval of our methods.

Some unexpected challenges came from the day to day interactions with their team. Often these interactions involved phone conferences, but habitually meeting invitations would be unacknowledged. On then other hand, our Finance partners had a habit of showing up to meetings they were not invited to, so we had to be ready to discuss LTV at any time. Straight answers were also not always simple to come by. For example, once to the question "Is this how we should be handling this expense category?" the reply was "What would happen if we lost all our customers?"

In the end the process resembled a Poisson process with a low probability of success. It was simply a matter of trying again and again until they (somewhat accidentally) said yes. As we found out later, Finance finally approved our formula because they did not think the LTV project would actually get finished and that if the LTV project were to be finished it would not be taken seriously.

The long process of working with Finance did have beneficial results. We had to think very carefully about how we were evaluating customers, and we had a much more robust formula at the end. From our experience and personal communication, any LTV project will require close and often contentious work with Finance.

Information Technology Department

A commonplace challenge to data mining projects is in the interaction with the information technology department. At LTC IT department's motto was "we will get you anything you want, just tell us what columns you want in your flat-file extract". Having another department producing production programming that would affect the data warehouse was a new idea to them.

We spent several months discussing the protocols of us working closely with IT programmers and establishing project resources (which included a very small disk space requirements on a server with spare capacity), only to have the whole plan shot down by IT management. The reason given was that the systems programming involved a model, and the IT department was not capable of handling models – only Marketing was. The result was that IT would drop off a data file and later pick up another file to load, but we would have to do all the programming in between.

At the end, this was a beneficial result, giving us necessary control over the output. However, the route did seem unnecessarily unpleasant. Managing IT's issues was primarily a matter of patience and flexibility.

"We Have to Stop This!": Small Influential Business Unit (SIBU)

SIBU was one of the groups we needed buy-in from. SIBU was responsible for a potentially highly profitable future line of business, and had identified a small group of current customers that they thought would be good targets for the new services. Because they were expected to be highly profitable in the future, SIBU had tremendous influence within LTC and their buy-off was needed for major projects.

SIBU's initial reaction to the project was absolute horror. Some of "their" customers might get poor scores! First, SIBU insisted that none of their potential customers get scored at all. They demurred when we pointed out this would mean essentially excluding them from all regular marketing efforts. SIBU's next idea was to stop the project completely.

We quickly realized that if we did not get the problem solved right then and there, the project would be dead. The solution to the challenge was the addition of a set of scores

(A/B/C/D/E) that was based on a special usage formula that SIBU provided.

As it turned out, by the time LTV went into production business had changed and the SIBU scores were irrelevant; all customers received a ‘C’. However, the scoring provided us the buy-in from SIBU necessary to complete the project without changing the core system.

Validating LTV

Ideally we would have validated the LTV scores by going back five years, scoring the customer base then, and then checking to compare our predictions versus reality. This was not possible. At LTC we only had 13 months of history available to us. In particular, we would not be able to test the critical assumption of the LTV system, that after an initial adjustment period a customer’s current behavior would on average be the customer’s future behavior.

What we did instead was to validate the decisions that the LTV system was indicating we should make. For example, the LTV system was indicating that a substantial population of customers was unprofitable because of off-network changes. LTC had to pay competing telecommunications companies fees to support the usage of LTC customers. To validate the LTV system we got the individual customer’s monthly usage, matched with our contract data to calculate how much each month we paid in fees to our competition, and could identify that we were losing substantial amounts of money keeping these customers as customers. This justified programs that gave substantial bonuses to customers to move to a plan where they paid for off-network usage.

There was yet another LTV project at LTC that failed this validation test. A European Consulting Company (ECC) had been brought in to do LTV in parallel with us. The ECC system relied heavily on network charges to calculate the monthly profit/loss on customers. Like our LTV system had identified off-network usage as an issue the ECC system identified large on-network usage as an issue, and ECC crafted a program to force migrate (i.e., ‘fire’) customers with high usage on family plans.

As the program managers drilled down into the data they became very dubious about the ECC program. They were targeting customers that were only using 50% of their allotted minutes, and this hardly seemed like excessive usage to the program managers. We found out about the ECC program when the program managers came to us for advice on the LTV scores of the targeted customers. The LTV scores showed no real difference between the targeted customers and other customers and this information was used to kill the program. This was the only time in my professional experience that I have seen program managers happy to have data kill a program.

AFTER PRODUCTION

Once the LTV system went into production, a new set of issues arose.

Customer Care

The first challenge after production was an unexpected demand to explain of individual scores. Fortunately the simple, modular nature of our LTV formula enabled quick and believable answers to all of these questions. For example the question “Why does Mr. Jones have high revenue and such a bad LTV score?” was commonly answered with “Mr. Jones had not paid us in X months”. (Surprisingly, the customer care system did not take payment history into account when handing out equipment). The ability to quickly provide clear, convincing answers to valuation questions gave the project a tremendous amount of credibility in the enterprise.

New Projects

Early into production the partners in the business and finance departments wanted different versions of the LTV scores. For example, they wanted either bad debt or off-network charges excluded from the equation. Because of the simple, modular nature of the formula these types of requests were feasible.

The LTV project became a springboard to other projects. For instance, when an outside consultancy group prioritized the items in LTC’s marketing budget based on local markets. The prioritization failed for several reasons, including: 1) it was a black box in that few knew its methodology 2) internal groups could not modify the results to produce their own analysis and 3) the prioritization only covered half of the markets. Producing a new prioritization with LTV was straightforward. Bad debt and off-network charges for example could be changed in order to show what could happen with tighter controls and better infrastructure. It was possible to deliver special LTV analysis that only focused on new customers so LTC cold see where to allocate acquisition dollars.

The only drawback of all this was that we had to do all the analysis. We only published the final results, and not all the intermediate quantities. This is something we would change if we could do the project again. Our business and finance partners would still look to us for guidance about LTV, but we could have them do the most of the work.

Regional Marketing Managers

After the project was put into production and we were educating LTC on the benefits and usage of LTV data, we ran into a substantial and justified conflict with some of the regional managers.

The issue was that LTC had expanded into areas ahead of LTC’s ability to profitably support the areas: build the customer base first, and then put in the infrastructure. The regional managers in these expansion areas were drastically affected by LTV scores. This effect was on a personal level: the manager’s abilities to meet their personal goals, and get their yearly bonus, were strongly effected.

We never got a full solution; the issue was still being discussed when we left the company. We were able to create partial solutions. Because of the modular nature of the LTV calculations, we were able to create an adjusted LTV that worked for these regions.

CONCLUSIONS

The LTV project was completed successfully. In addition to the company benefits, there were substantial career benefits: our department became known and respected as the 'LTV Department', and we gained tremendous credibility in all our other projects because of this.

Looking back, the key ingredients to the success were

1) The project had solid value with a rock-solid analytic base. The substance of the project really does matter. Because we had solid analytics the team believed in the project and we were able to defend the project against criticism.
2) The project had a high-level sponsor that was willing to go out on a limb for the project. Needing an executive sponsor is a truism that is actually true.
3) The team could make decisions about the project without having to go back to the sponsor. Many times (most notably with SIBU) we were negotiating with other business units, and the sponsor had to trust us to get it right.
4) We designed the end result to be usable.
5) The core design team and implementation team were the same. A hand-off between design and implementation is a natural place for projects to die.

After the LTV project, Finance initiated an LTV-like project for Activity-Based Costing. This was so that LTC could get a handle on its costs at a very low level, which was something LTC desperately needed. However, what LTC Finance did was to first hire a consulting group for a year of design work. The consultant group had endless meetings with every group in the company, and eventually produced a massive design document. The design spec went to IT, IT replied the project would take $16M to build, and the project was shelved.

Design teams need to understand what the implementation issues are and implementation teams need to understand what the design priorities are. If the teams do not share the same core then they need to be able to work very closely.

What We Would Do Differently

There were some things that did not go well. Topics worth further experimenting with in future projects include:

1) Put together a simple reporting engine running off of our desktops first. A system like this could have spotted the regional problem.
2) Publish all the calculational components of the LTV system, in order for users of the system to be able to customize the results.

ACKNOWLEDGMENTS

Our thanks go to Brendan Kitts for his preliminary review and feedback.

REFERENCES

[1] D. R. Cox and D. Oakes. Analysis of Survival Data. Chapman and Hall, London, (1984).

[2] C. Cunningham, I. Song, and P. Chen. Data warehouse design to support customer relationship management analyses. ACM CIKM 2005, DOLAP Workshop, ISBN:1-58113-977-2, (2005), 14-22

[3] F. R. Dwyer. Customer Lifetime Valuation for Marketing Decision Making. Journal of Direct Marketing, 11, 4 (1997), pp 6-13

[4] H. Hwang, T. Jung, and E. Suh. An LTV model and customer segmentation based on customer value: a case study on the wireless telecommunication industry. Expert Systems with Applications, 26, (2004), 181-188.

[5] S. Rosset, and E. Neumann. Integrating Customer Value Considerations into Predictive Modeling. IEEE ICDM 2003, ISBN:0-7695-1978-4, (2003), 283.

[6] S. Rosset, E. Neumann, U. Eick, and N. Vatnik. Customer Lifetime Value Models for Decision Support. Data Mining and Knowledge Discovery, 7, 3 (July 2003), 321-339.

[7] S. Rosset, E. Neumann, U. Eick, N. Vatnik, and Y. Idan. Customer lifetime value modeling and its use for customer retention planning. ACM KDD 2002, (July 2002), 332-340.

Survival analysis models to estimate customer lifetime value

Silvia Figini[1] Paolo Giudici[1] Claudia Gagliano[2] Daniela Polla[2]
[1]*Data Mining Laboratory, University of Pavia*
silvia.figini[ατ]eco.unipv.it;
[2]*Sky Italy, Milan*

Abstract

We consider the problem of estimating customer lifetime value of customers, when a large number of features are present in the data. In order to measure lifetime value we use survival analysis models to estimate customer tenure. One of the most popular methods in the analysis of survival data is the comparison of hazard and survival functions, for different classes of patients and /or treatments Such comparison, along with Cox proportional hazard model, is well suited to infer the effect of covariates on survival time.

In the paper we show how these ideas and methods can be adapted to a different environment, the estimation of customer's life cycles. In such a context, a number of data mining modelling challenges arise. For example, differently from a patient, a client that is lost can come back.

We will show how our approach performs, and compare it with classical churn models on a real case study, based on data from a media service company that aims to predict churn behaviours, in order to entertain appropriate retention actions.

1. Introduction

In the global competitive marketplace, businesses regardless of size are beginning to realize that one of the keys to profitable growth is establishing and nurturing a one-on-one relationship with the customer. Businesses now realize that retaining and growing existing customers is much more cost effective than focusing primarily on adding new customers. To this end, techniques for customer relationship management (CRM) are being designed, developed and implemented. These techniques should help businesses understand customer needs and spending patterns, and help develop targeted promotions which are not only better tailored for each customer, but are also profitable to the business in the long run. Customer lifetime value (LTV), which measures the profit generating potential, or value, of a customer, is increasingly being considered a touchstone for administering the CRM process. This in order to provide attractive benefits and retain high-value customers, while maximizing profits from a business standpoint. Robust and accurate techniques for modelling LTV are essential in order to facilitate CRM via LTV. A customer LTV model needs to be explained and understood to a large degree before it can be adopted to facilitate CRM. LTV is usually considered to be composed of two independent components: tenure and value. Though modelling the value (or equivalently, profit) component of LTV, (which takes into account revenue, fixed and variable costs), is a challenge in itself, our experience has revealed that finance departments, to a large degree, well manage this aspect. Therefore, in this paper, our focus will mainly be on modelling tenure rather than value.

A variety of statistical techniques arising from medical survival analysis (see e.g., [3]) can be applied to tenure modeling. We look at tenure prediction using classical survival analysis and compare it with data mining techniques that use decision tree and logistic regression. In our business problem the survival analysis approach performs better with respect to a classical data mining predictive model for churn reduction (e.g. based on regression or tree models, see Section 3). In fact, the key challenge of LTV prediction is the production of segment-specific estimated tenures, for each customer with a given service supplier, based on the usage, revenue, and sales profiles contained in company databases. The tenure prediction models we have developed generate, for a given customer i, a hazard curve or a hazard function, that indicates the probability $h_i(t)$ of cancellation at a given time t in the future. A hazard curve can be converted to a survival curve or to a survival function which plots the probability $S_i(t)$ of "survival" (non-cancellation) at any time t, given that customer I was "alive" (active) at time (t-1), i.e., $S_i(t)=S_i(t-1) \times [1-h_i(t)]$ with $S_i(1)=1$. Once a survival curve for a customer is available, LTV for that specific customer i can be computed as:

$$\text{LTV} = \sum_{t=1}^{T} S_i(t) \times v_i(t) \text{ , (1.1)}$$

where $v_i(t)$ is the expected value of customer i at time t and T is the maximum time period under consideration. The approach to LTV computation provides customer specific estimates (as opposed to average estimates) of the total expected future (as opposed to past) profit based on customer behaviour and usage patterns. In the realm of CRM, modelling customer LTV has a wide range of applications including:

- Evaluating the returns of the investments in special offers and services.
- Targeting and managing unprofitable customers.
- Designing marketing campaigns and promotional efforts
- Sizing and planning for future market opportunities

Some of these applications would use a single LTV score computed for every customer. Other applications require a separation of the tenure and value component for effective implementation, while even others would use either the tenure or value and ignore the other component. In almost all cases, business analysts who use LTV are most comfortable when the predicted LTV score and/or hazard can be explained in intuitive terms.

Our case study concerns a media service company. For non disclosure reasons in this paper we cannot give accurate statements and information about the company whose data we have analysed; we shall instead use general statements and normalised figures; each table and figure reported in the paper is determined on a random sample from the real data; the company will be referred to as "the company".

The main objective of such a company is to maintain its customers, in an increasingly competitive market; and to evaluate the lifetime value of such customers, to carefully design appropriate marketing actions.

2. Definition of churn

The company is such that most of its sales of services are arranged through a yearly contract, that allows buying different "packages" of services at different costs. The contract of each customer with the company is thus renewed yearly. If the client does not withdraw, the contract is renewed automatically. Otherwise the client churns. In the company there are three types of churn events: people that withdraw from their contract in due time (i.e. less than 60 days before the due date); people that withdraw from their contracts overtime (i.e. more than 60 days before the due date); people that withdraw without giving notice, as is the case of bad payers. Correspondingly, the company assigns two different churn states: an 'EXIT' state to the first two classes of cutomers; and a 'SUSPENSION' state to the third.

Concerning the causes of churn, it is possible to identify a number of components that can generate such a behaviour:

- a static component, determined by the characteristics of the customers and the type/subject of contracts;
- a dynamic component, that encloses trend and the contacts of the clients with the call center of the company;
- a seasonal part, tied to the period of subscription of the contract;
- external factors, that include the course of the markets and of the competitors.

Currently the company uses a data mining model that gives, for each customer, a probability of churn (score). The goal for the company is to identify customers that are likely to leave and join a competitor. This objective is well perceived by the top management, which considers lowering churn one of the key targets of the company.

3. Traditional data mining churn models

The churn model used in the company to predict churn is currently a classification tree. Tree models can be defined as a recursive procedure, through which a set of n statistical units is progressively divided in groups, according to a divisive rule which aims to maximize a homogeneity or purity measure of the response variable in each of the obtained groups. At each step of the procedure, a divisive rule is specified by: the choice of an explanatory variable to split; the choice of a splitting rule for such variable, which establishes how to partition the observations.

The main result of a tree model is a final partition of the observations: to achieve this it is necessary to specify stopping criteria for the divisive process. Suppose that a final partition has been reached, consisting of *g* groups (g < n). Then, for any given observation response variable observation y_i, a regression tree produces a fitted value $\hat{y}_i$ which is equal to the mean response value of the group to which the observation i belongs. Let *m* be such group; formally we have that:

$$\hat{y}_i = \frac{\sum_{l=1}^{n_m} y_{lm}}{n_m} \quad (3.1)$$

For a classification tree, instead, as seen for logistic regression, fitted values are given in terms of fitted probabilities of affiliation to a single group. Suppose only two classes are possible (binary classification); the fitted success probability is therefore:

$$\pi_i = \frac{\sum_{l=1}^{n_m} y_{lm}}{n_m} \quad (3.2)$$

where the observations y_{lm} can take either the value 0 or 1, and, therefore, the fitted probability corresponds to the observed proportion of successes in the group m. Notice that both $\hat{y}_i$ and π_i are constant for all the observations in the group. From the point of view of the visual display of the results, the output of the analysis is usually represented through a tree, that is very similar, in structure, to a tree of hierarchical clustering [4].

Tree models may show problems in time-dependent applications, such as churn applications. For example, for the company data at hand, even the best tree model classifies as customers most at risk nearly all those customers whose deadline to withdraw falls in the months immediately after the application of the model. Such a prediction can hardly be of effective support to decide which customers to target for retention activities aimed at reducing churn.

In fact, identifying as customers at risk the 80-90% of customers whose deadline date falls in the following months (as the best performing tree models indeed do) leads to a target customer base which it too large and to expensive to make actions upon.

In any case, we now describe how we have actually applied classification tree models in our context. The response variable, used as a dependent variable to build predictive models, includes two different types of customers: those who during the survey are active and those, instead, who regularly cancelled their subscription (EXIT status). We remark that, due to the different time nature of the withdrawal, SUSPENSION status customers cannot be simply included in the analysis but, rather, require a specific treatment. They can instead be merged in the survival analysis context (as in Section 5).

We remark that the target variable has been observed 3 months after the extraction of the data set used for the model implementation phase, so to verify correctly the effectiveness and predictive power of the models themselves.

Concerning explanatory variables, the available variables employed were taken from different databases used inside the company, which contained, respectively: socio-demographic information about the customers; information about their contractual situation and about its changes in time; information about contacting the customers (through the call centre, promotion campaigns, etc) and, finally, geo-marketing information (divided into census, municipalities and larger geographical sections).

The variables regarding customers contain demographic information (age, gender, marital status, location, number of children, job and degree) and other information about customer descriptive characteristics: hobbies, pc possession at home, address changes.

The variables regarding the contract contain information about its chronology (signing date and starting date, time left before expiration date), its value (fees and options) at the beginning and at the end of the survey period, about equipments needed to use services (if they are rented, leased or purchased by the customer) and flags which indicate if the customer has already had an active, cancelled or suspended contract.

There are also information about invoicing (invoice amount compared to different period of time – 2, 4, 8, 12 months). The variables regarding payment conditions include information about the type of payment of the monthly subscription (postal bulletin, account charge, credit card), with other info about the changes of the type of payment. The data set used for the analysis also includes variables which give info about the type of the services bought, about the purchased options, and about specific ad-hoc purchases, such as number and total amount of specific purchases during the last month and the last 2 months.

The variables regarding contacts with the customer contain information about any type of contact between the customer and the company (mostly through calls to the call centre). They include many types of calling categories (and relatives subcategories). They also include information about the number of questions made by every customer and temporal information, such as the number of calls made during the last month, the last two months and so on.

Finally, geomarketing variables are present at large, and a great amount of work has involved their pre-processing and definition.

Regardless of their provenience, all variables have gone through a pre-processing feature selection step aimed at reducing their very large number (equal to

606). Such step has been perfomed using a combination of wrapping and filter techniques, going from dimensionality reduction to association measure ranking.

To increase predictive model performance we have also decided to employ a classification tree, in a purely explorative point of view (see [4]. For example, the continuous variables were discretized in classes, following the subdivision generated by the tree. In this way the allocation of levels was not done following quantities such as deciles or quartiles, usually used for this purpose, but using classes which best predict the variable target.

We have compared the predictive performance of the best classification tree with logistic regression models. More generally, all created models were evaluated on the basis of a test sample (by definition not included in the training phase) and classified in terms of predictive accuracy with respect to the actual response values in it. In business terms, predictive accuracy means being able to identify correctly those individuals which will become really churner during the valuation phase (correct identification).

Evaluation was made using a confusion, or cross validation matrix. On the basis of our results: the confusion matrix of the predictive tree, built with a CART method, says that calling 18.5% of the sample customers (25499) during a year, we can identify 41.7% of those who will actually become EXIT (6906). In the list of the reachables (25499) we can find 27.1% of real EXIT, against 12% when considering the whole sample. This means the model let us identify the customers at risk with an increased accuracy equal to 2.26 if compared to a random extraction (27.1/12=2.26).

The cross validation matrix obtained from a Chaid tree says that calling 26.3% of the sample customers during a year, we can identify 51.9% of those who will actually become EXIT. In the list of the reachables we can find 23.7% of real EXIT, against 12% if considering the whole sample. This means the model let us identify the customers at risk with an increased accuracy equal to 2 if compared to a random extraction (23.7/12=2). So these two types of trees have a good lift, that is, are able to predict a good churn rate and the CART one shows a greater capability than Chaid tree.

However, there is the problem of the excessive influence of the contract deadline. The two types of trees predict that 90% of customers whose deadline is in April is at risk. If we consider that the variable target was built gathering data of February, the customers whose term is in April and have to regularly unsubscribe within the 60 days allowed, must become EXIT in February. Therefore, despite their good predictive capability, these models are useless for marketing actions, as a very simple model based on customer's deadlines will perform as well.

4. Weaknesses of traditional data mining models

The use of new methods is necessary to obtain a predictive tool which is able to consider the fact that churn data is ordered in calendar time. To summarise, we can sum up at least four main weaknesses of traditional models in our set-up, which are all related to time-dependence:

- excessive influence of the contract deadline date, also shown by a high association of some variables in the database with the month of deadline;
- redundance of information: the database contains variables which gives redundant information; as these variables are time dependent, they may induce biased effects in the final estimates
- presence of fragmentary information (variables regarding personal data, demands, pay per view activities, all with high lack of data), depending on the measurement time;
- excessive weight of the different temporal perspectives (the method used to build predictive models cannot catch this temporal dimension).

The previous points explain why we decided to look for a novel and different methodology to predict churn. Last, and perhaps most important, from the company viewppoint, classical models have led to a too large campaign outreach, whose budget expense is indeed too large too support. This is the main reason why the company has supported our choice to look for a new model.

We have chosen the survival analysis approach: this method was born in the medical world, but in the company we applied it in a novel way to predict churn behaviour.

5. Survival analysis models to estimate churn

We now turn our attention towards the application of methodologies aimed at modelling survival risks. In our case study the risk concerns the value that derives from the loss of a customer. The objective is to determine which combination of covariates affect the risk function, studying specifically the characteristics and the relation with the probability of survival for every customer.

Survival analysis is concerned with studying the time between entry to a study and a subsequent event (churn). All of the standard approaches to survival analysis are probabilistic or stochastic. That is, the times at which events occur are assumed to be realizations of some random processes. It follows that T, the event time for some particular individual, is a random variable having a probability distribution.

A useful, model-free approach for all random variables is nonparametric, that is, using the cumulative distribution function. The cumulative distribution function of a variable T, denoted by F(t), is a function that tell us the probability that the variable will be less than or equal to any value t that we choose. Thus, F(t) = P{T≤t}. If we know the value of F for every value of t, then we know all there is to know about the distribution of T. In survival analysis it is more common to work with a closely related function called the survivor function defined as S(t) = P{T>t} = 1 - F(t). If the event of interest is a death (or, equivalently, a churn) the survivor function gives the probability of surviving beyond t. Because S is a probability we know that it is bounded by 0 and 1 and because T cannot be negative, we know that S(0) =1. Finally, as t gets larger, S never increases. Often the objective is to compare survivor functions for different subgroups in a sample (clusters, regions…). If the survivor function for one group is always higher than the survivor function for another group, then the first group clearly lives longer than the second group.

When variables are continuous, another common way of describing their probability distributions is the probability density function. This function is defined as:

$$\mathrm{f(t)} = \frac{dF(t)}{dt} = -\frac{dS(t)}{dt} \quad (5.1).$$

That is, the probability density function is just the derivative or slope of the cumulative distribution function. For continuous survival data, the hazard function is actually more popular than the probability density function as a way of describing distributions. The hazard function is defined as :

$$\mathrm{h(t)} = \lim\ \varepsilon t \rightarrow 0 \ \frac{\Pr\{t \leq T < t + \varepsilon t \mid T \geq t\}}{\varepsilon t} \ (5.2)\ .$$

The aim of the definition is to quantify the instantaneous risk that an event will occur at time t. Since time is continuous, the probability that an event will occur at exactly time t is necessarily 0. But we can talk about the probability that an event occurs in the small interval between t and t+ εt and we also want to make this probability conditional on the individual surviving to time t. For this formulation the hazard function is sometimes described as a conditional density and, when events are repeatable, the hazard function is often referred to as the intensity function. The survival function, the probability density function and the hazard function are equivalent ways of describing a continuous probability distribution. Another formula expresses the hazard in terms of the probability density function :

$$\mathrm{h(t)} = \frac{f(t)}{S(t)} \ (5.3)$$

and together equations (5.3) and (5.1) imply that

$$\mathrm{h(t)} = -\frac{d}{dt} \log \mathrm{S(t)} \ (5.5).$$

Integrating both sides of equation (5.5) gives an expression for the survival function in terms of the hazard function:

$$\mathrm{S(t)} = \exp\ (- \int_0^t h(u)du\ ) \quad (5.6).$$

With regard to numerical magnitude, the hazard is a dimensional quantity that has the form: number of events per interval of time. The interpretation of the hazard as the expected numbers of events in a one-unit interval of time makes sense when events are repeatable. The database available for our analysis contain information that can affect the distribution of the event time, as the demographic variables, variables about the contract, the payment, the contacts and geomarketing.

We remind the readers that the target variable has a temporal nature and, for this reason, it is preferable to build predictive models through survival analysis.

The actaul advantages of using a survival analysis approach with respect to a traditional one can be reassumed in following:

- to correctly align the customers regarding their cycle of life;
- to analyze the real behaviour of the customers churn, without having to distinguish between EXIT and SUSPENSION payers.

In order to build a survival analysis model, we have constructed two variables: one variable of status (distinguish between active and non active customers) and one of duration (indicator of customer seniority) . The first step in the analysis of survival data (for the descriptive study) consists in a plot of the survival function and the risk. The survival function is estimated through the methodology of Kaplan Meier[6]. The Kaplan Meier estimator is the most widely used method for estimating a survival function and it is based on a nonparametric maximum likelihood estimator. When there are non censored data the KM estimator is just the sample proportion of observations with event times greater than t. The situation is also quite simple in the case of single right censoring, that is, when all the censored cases are censored at the same time c and all the observed event time are less then c. In that case, for all $t \leq c$ the KM estimator is still the sample proportion of observations with events time greater then t. For t>c the estimator is undefined. Things get more complicated when some censoring times are smaller then some event times. In that instance, the observed proportion of cases with event times greater then t can be biased downward because cases that are censored before t may, in fact, have "died" before t without our knowledge. The solution is as follows. Suppose there are K distinct event times, $t_1<t_2<\ldots<t_k$. At each time t_j there are n_j individuals who are said to be at risk of an event. At risk means they have not experienced an event not have they been censored prior to time t_j . If any cases are censored at exactly t_j , there are also considered to be at risk at t_j . Let d_j be the number of individuals who die at time t_j . The KM estimator is defined as

$$\hat{S}(t) = \prod_{j:t_j \leq t} [1 - \frac{d_j}{n_j}] \text{ (5.7) for } t_1 \leq t \leq t_k .$$

This formula says that, for a given time t, take all the event times that are less than or equal to t. For each of those event times, compute the quantity in brackets, which can be interpreted as the conditional probability of surviving to time t_{j+1} , given that one has survived to time t_j . Then multiply all of these survival probability together.

We now consider our application to the company data. Figure 1 shows the survival function for the whole customer database.

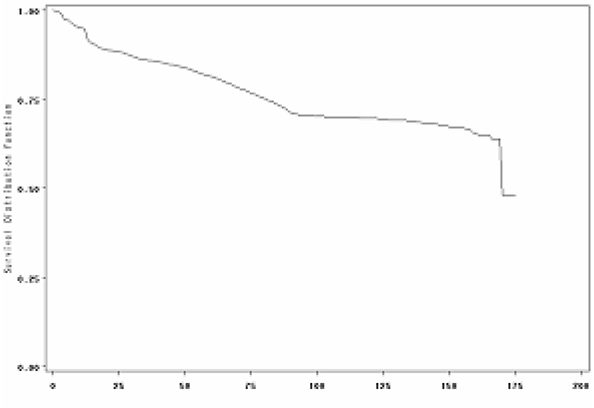


Figure 1: Descriptive Survival function

From Figure 1 note that the survival function has varying slopes, corresponding to different periods. When the curve decreases rapidly we have time periods with high churn rates; when the curve decreases softly we have periods of "loyalty" We remark that the final jump is due to a distorsion caused by a few data, in the tail of the lifecycle distribution.

In Figure 2 we show the hazard function, that shows how the instantaneous risk rate varies in time.

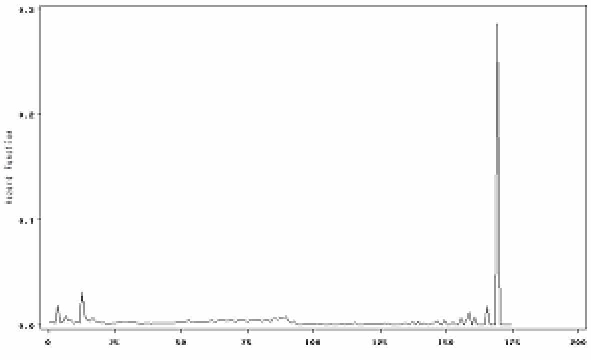


Figure 2: Hazard function

From Figure 2 we note two peaks, corresponding to months 4 and 12, the most risky ones. Note that the risk rate is otherwise kept almost constant along the lifecycle. Of course there is a peak in the end corresponding to what observed in Figure 1.

A very useful information, in business terms, is the calculation of the life expectancy of the customers. This can be obtained as a sum over all observed event times:

$\hat{S}(t_{(j)})(t_{(j)} - t_{(j-1)})$, where $\hat{S}(t_{(j)})$ is the estimate of the survival function at the j-th event time, obtained using Kaplan Meier method, and *t* is a duration indicator. ($t_{(0)}$ is by assumption equal to 0). We remark that life expectancy tends to be underestimated if most observed event types are censored (i.e., no more observable).

We now move to the comparison of hazard and/or survival curves, according to the available covariates [2]. Figure 3 shows an example of such comparison, on the basis of the geographical information concerning the region where the customer lives.

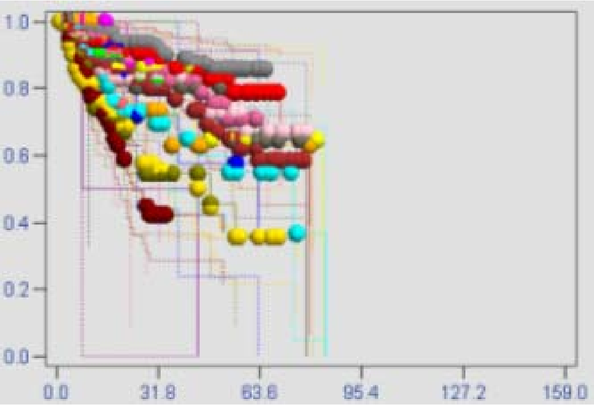


Figure 3: Geographic survival function

In Figure 3 the x-axis indicate lifetime and the y-axis the survival probabilities. From figure 3 one can appreciate significant differences in survival curves, due to the geographical factor. These differences have been further confirmed by a formal test of hypotheses (Scheffè multiple comparison test).

Conclusions similar to those in Figure 3 can be obtained for all explanatory variables; in our experience this represents a great wealth for business usages.

We now move to the building of a full predictive model. We have chosen to implement Cox's model [3]. Cox made two significant innovations. First he proposed a model that is a proportional hazards model. Second he proposed a new estimation method that was later named partial likelihood or more accurately, maximum partial likelihood. We will start with the basic model that does not include time-dependent covariate or non proportional hazards. The model is usually written as:

$$\mathrm{h(t_{ij})} = \mathrm{h}_0(t_j)\exp\lfloor \beta_1 X_{1ij} + \beta_2 X_{2ij} + \cdots + \beta_P X_{Pij} \rfloor$$

(5.8)

Equation 5.8 says that the hazard for individual i at time t is the product of two factors: a baseline hazard function that is left unspecified, and a linear combination of a set of p fixed covariates, which is then exponentiated. The baseline function can be regarded as the hazard function for an individual whose covariates all have values 0. The model is called proportional hazard model because the hazard for any individual is a fixed proportion of the hazard for any other individual. To see this, take the ratio of the hazards for two individuals I and j:

$$\frac{h_i(t)}{h_j(t)} = \exp \{ \beta_1 (x_{i1} - x_{j1}) + \ldots + \beta_p (x_{ip} - x_{jp}) \}$$

(5.9)

What is important about this equation is that the baseline cancels out of the numerator and denominator. As a result, the ratio of the hazards is constant over time.

In Cox model buiding the objective is to identify the variables that are more associated with the churn event. This implies that a model selection exercise, aimed at choosing the statistical model that best fits the data, is to be carried out. The statistical literature presents many references for model selection (see e.g. [4]).

Most models are based on the comparison of model scores. The main score functions to evaluate models are related to the Kullback-Leibler principle. This occurs for criteria that penalize for model complexity, such as AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion). To find the AIC in 1974 Akaike formulated the idea that (i) the parametric model is estimated using the method of maximum likelihood and (ii) the parametric family specified contains the unknown distribution f(x) as a particular case. He therefore defined a function that assigns a score to each model by taking a function of the Kullback-Leibler sample discrepancy. In formal terms the AIC criterion is defined by the following equation:

$$AIC = -2\log L(\hat{\vartheta}; x_1, \ldots, x_n) + 2q \quad (5.10)$$

wherw $\log L(\hat{\vartheta}; x_1, \ldots, x_n)$ is the logarithm of the likelihood function, calculated in the maximum likelihood parameter estimate and q is the number of parameters of the model. Notice that the AIC score essentially penalises the log-likelihood score with a term that increases linearly with model complexity.

The AIC criterion is based on the implicit assumption that q remains constant when the size of the sample increases. However this assumption is not always valid and therefore the AIC criterion does not lead to a consistent estimate of the dimension of the unknown model.

An alternative, and consistent, scoring function is the BIC criterion, also called SBC. It was formulated by Schwarz (1978) and is defined by the following expression:

$$BIC = -2\log L\left(\hat{\vartheta}; x_1, \ldots, x_n\right) + q\log(n) \quad (5.11)$$

As is easily seen the BIC differs from the AIC only in the second part which now also depends on the sample size n. Compared to the AIC, when n increases the BIC favours simpler models. As n gets large, the first term (linear in n) will dominate the second term (logarithmic in n). This corresponds to the fact that, for a large n, the variance term in the mean squared error expression tends to be negligible. We also point out that, despite the superficial similarity between the AIC and the BIC, the first is usually justified by resorting to classical asymptotic arguments, while the second by appealing to the Bayesian framework.

To conclude, the scoring function criteria for selecting models which we have examined are easy to calculate and lead to a total ordering of the models. From most statistical packages we can get the AIC and BIC scores for all the models considered. A further advantage of these criteria is that they can be used also to compare non-nested models and, more generally, models that do not belong to the same class (for instance a probabilistic neural network and a linear regression model).

However, the limit of these criteria is the lack of a threshold, as well the difficult interpretability of their measurement scale. In other words, it is not easy to determine if the difference between two models is significant or not, and how it compares to another difference [4].

In our case we have compared all models and chose the one with the lowest value of AIC and BIC. The obtained model presents a good fit, as shown in Table 1 below.

Model Fit Statistics		
Criterion	Without Covariates	With Covariates
-2 LOG L	1124678.6	943660.62
AIC	1124678.6	943788.62
BIC	1124678.6	944347.95

Table 1: Goodness of fit statistics

From Table 1 note that both AIC and BIC present lower values with the inclusion of covariates: this means that adding covariates lead to a better fit. As well known, the BIC presents higher values due to its penalty term.

Log likelihood comparison can be formally embedded into an overall statistical test, such as Score, Wald or the likelihood ratio test [6]. These test compares the null assumption of no covariate effect against the alternative that at least one is different from zero. Table 2 shows the results of all three tests.

Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	181018.0	64	<.0001
Score	344161.2	64	<.0001
Wald	135991.4	64	<.0001

Table 2: Measures of significance of the parameters

From Table 2 we have further evidence on the goodness of fit of the chosen model, using either test. We underline that the final model contains 64 explanatory variables, from the starting 606.

We remark that, on each variable parameter, it is possible to undertake a further variable-specific test, e.g. based on the Score or Wald test. We do not presents such results, for lack of space and non-disclosure issues.

Indeed the chosen model contains too many variables for a simple interpretation. We have therefore chosen to simplify further the model, by means of a stepwise model selection procedure, based on the likelihood ratio test, and a significancy level of 0,05, starting from the model in Tables 1 and 2.

The result of the procedure is a set of about twenty explanatory variables. Such variables can be grouped in three main categories, according to the sign of their association with the churn rate, represented by the hazard ratio. We cannot disclose which are the signs of the effects of those variables, however we point out to the reader that the significant variables concern the wealth of the geographic region, the quality of the call center service, the sales channel, the number of technical problems, the cost of the service bought, the status of the payment method.

More precisely, to calcolate the previous associations we have considered the values of the hazard ratio under different covariate values. For example, for the variables indicating number of technical problems we compare the hazard function for those that have called at least once with those that have not made such calls.

A very important remark is that Cox model generates survival functions that are adjusted for covariate values. More precisely, the survival function is computed according to the following [see 7]:

$$S(t,X)=S_0(t)\exp(\Sigma\beta_i X_i) \quad (5.12)$$

Figure 4 below shows a comparison between the survival curve obtained without covariates (the baseline, as in Figure 1) and the same curve adjusted for the presence of covariates (obtained as in equation 5.12).

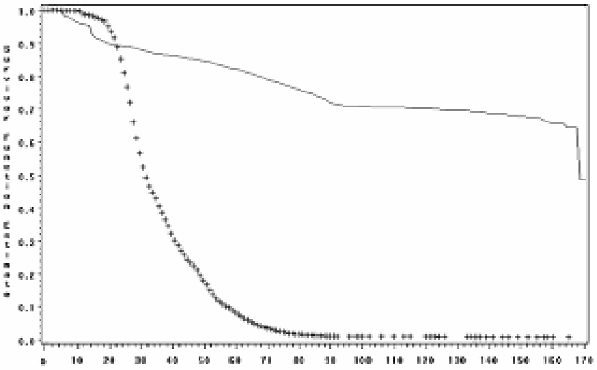


Figure 4: Comparison between survival functions

Figure 4 shows that covariates affect considerably survival times: up to two years of lifetime, the Cox survival curve (described by the symbols "+") is greater with respect to the baseline (described by the continuous curve). After such period the survival probability declines abruptly and turns out to be much lower for the remaining lifespan.

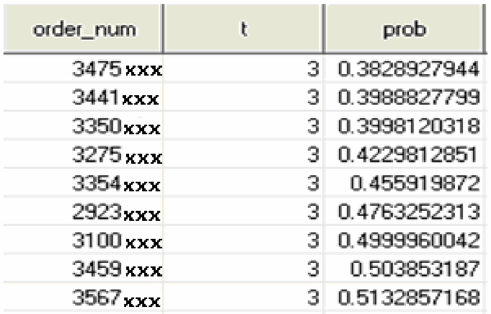
Once a Cox model has been fitted, it is advisable to produce diagnostic statistics, based on the analysis of residuals, to verify if the hypotheses underlying the model are correct. In our case they were found to be correct, so we could proceed with the predictive stage.

In the prediction step the goodness of the model will be evaluated in terms of predictive accuracy.

We have first split the dataset in the two usual subsets: training and test. Both samples have been proportionally sampled, with respect to the status variable. All sampled data contain information on all final chosen explanatory variables (about twenty). The response (status) variable has been built reporting customer status at the end of june 2005. Notice that this date is two months later than what done for classical models, this is motivated by the simple fact that survival analysis modelling was done afterwards, thus data have been updated in the customer database.

We remark that the training set is of course much larger than the validation set. Furthermore, differently from what occurs with classical data mining models, here we have considered, as churners, both EXIT and SUSPENSION status customers.

We also remark that survival analysis models allow the prediction of a future curve, not of a single point; indeed a three months ahead prediction point was chosen for marketing purposes (it is in line with early warning needs). Table 3 shows an extract of the 3 months ahead survival probabilities estimated by the model.



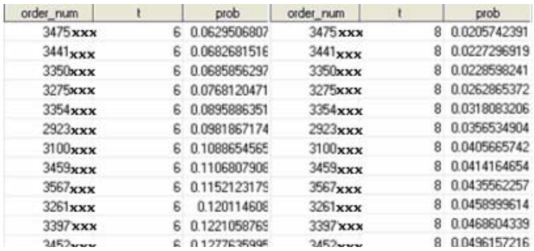
order_num	t	prob
3475xxx	3	0.3828927944
3441xxx	3	0.3988827799
3350xxx	3	0.3998120318
3275xxx	3	0.4229812851
3354xxx	3	0.455919872
2923xxx	3	0.4763252313
3100xxx	3	0.4999960042
3459xxx	3	0.503853187
3567xxx	3	0.5132857168

Table 3: Survival Probability Estimated for each customer, three monhths ahead of time

Table 3 shows, for each customer (one for each row) , their 3 months ahead survival probabilities.

As we said above, the model can give predictions for any k-month ahead period, although these may receive less marketing action priorities.

Table 4 below contains, for example, for the same clients as in Table 3, the predicted survival probabilities for 6 and 8 months ahead from the data extraction time (march 2005, therefore predictions for september and novembre 2005).



order_num	t	prob	order_num	t	prob
3475xxx	6	0.0629506807	3475xxx	8	0.0205742391
3441xxx	6	0.0682681516	3441xxx	8	0.0227296919
3350xxx	6	0.0685856297	3350xxx	8	0.0228598241
3275xxx	6	0.0768120471	3275xxx	8	0.0262865372
3354xxx	6	0.0895886351	3354xxx	8	0.0318083206
2923xxx	6	0.0981867174	2923xxx	8	0.0356534904
3100xxx	6	0.1088654565	3100xxx	8	0.0405665742
3459xxx	6	0.1106807908	3459xxx	8	0.0414164654
3567xxx	6	0.1152123179	3567xxx	8	0.0435562257
3261xxx	6	0.120114608	3261xxx	8	0.0458999614
3397xxx	6	0.1221058769	3397xxx	8	0.0468604339
3452xxx	6	0.1277635995	3452xxx	8	0.0496157216

Table 4: Survival Probability Estimated for each customer in different ahead times (6 months, 8 months)

From Table 4, note that the probability ranking do not change between 6 and 8 months (and do not w.r.t Table 3). This derives directly from the proportional hazard assumption.

In order to evaluate the predictive performance of the model, and compare it with classical data mining models, we have focused our attention to the 3 months ahead prediction.

Once survival probabilities have been calculated, we have deviced and implemented a procedure to build the confusion matrix and, correspondingly, the percentage of captured true churners of the model. We remark that this is indeed not a fair comparison as survival models predict more than a point; however the company wanted this comparison as well.

Table 5 contains the results of such comparison; in correspondence of each estimated probability decile, we report the percentage true churners in it (%captured).

decile	% captured
1	5.04
2	3.45
3	2.39
4	1.94
5	2.34
6	2.87
7	2.81
8	2.02
9	1.96
10	1.32

Table 5: captured response

From Table 5 note that, while in the first decile (that is, among the customers with the highest estimated churn probability) 5% of the clients are effective churners, the same percentage lowers down in susequent deciles, thus giving an overall picture of good performance of the model.

Indeed the lift of the model, as measured by the ratio between the captured true responses (model vs random) does not turn out to be substantially better with respect to what obtained with tree models.

However, we remark that, differently from what occurred with classical models, the customers with the highest estimated churn rate are now not necessarily those whose contract is close to the deadline. This is the most beneficial advantage of the survival analysis approach, that, in turn, leads to substantial gains in campaign costs.

A further advantage of the survival analysis approach lies in its immediate translation in terms of lifetime value analysis, as we shall see in the next Section.

6. Survival analysis models to estimate customer lifetime value

The aim of this last Section is to employ the results from survival analysis modelling to create a statistical model that allows to estimate the lifetime value of each customer, or, in a perhaps more useful aggregated analysis, for each segment of customers [5].

In other words, survival analysis is useful to quantify, in precise monetary terms, how much is gained or how much is lost by moving through different strata corresponding to different survival curve. For instance ,how much is gained/lost if 8% of the clients, say, switch from buying service A to buying service B.

Or, similarly, the relative gains when a certain percentage of clients change method of payment (e.g. moving between banking account, credit card or postal order).

In order to quantify gains and losses, a simple mesure is to calculate the area between the two corresponding survival curves, as shown in Figure 5 below. Suppose the two survival curves correspond to two different services bought, say Blue and Pink, corresponding to the colors of the two curves.

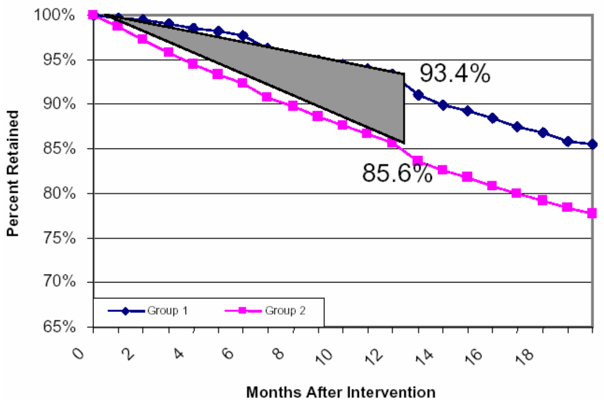


Figure 5: Evaluation of gain/losses by comparing survival curves

In order to determine exactly the area in Figure 5 we need to specify a temporal period ahead, e.g. 13 months in the Figure. In the Figure, the difference between survival probabilities after 13 months of life of the customers (e.g. 13 months since the first contact), is equal to 7.8 %. This value should be multiplied by the difference in business margin between the two methods of payment, as given, for example, by the difference in costs. Such costs can be described by a gain table, as in Table 6.

	PO	CC	RID
PO	▨	A	B
CC		▨	C
RID			▨

Table 6: relative gains between different methods of payment. PO = postal order; CC= payment through credit card; RID= payment through banking account

From Table 6, a value of A is the relative gain if the client switches from PO to CC and, similarly, B and C corresponds to relative gains switching from PO to RID and CC to RID.

In terms of Figure 5, if we assume that we start with an acquired client base of 1000 customers in both categories (product blue buyers and product pink buyers), the results say that, after 13 months we will remain with 934 blue and 856 pink. If the finance department tell us that product blue is worth 10$ and product pink 20$ we have that, after 13 months, we lose 660$ for blue churners and 2880 for pink churners. In other words, the priority of the marketing department should be to build targeted campaigns for pink product clients.

From a different perspective, if blue and pink correspond to two different selling channels of the same product, or to two different geographical areas, it is clear that the blue channel (or area) is much wiser in terms of customer retention. Often promotional campaigns are conducted looking only at increasing the customer base. Our results show that the number of captured clients should be traded with their survival or, better, lifetime value profile.

10. Conclusions

In the paper we have presented a comparison between classical and novel data mining techniques to predict rates of churn of customers. Our conclusions show that the novel approach we have proposed, based on survival analysis modelling, leads to more robust conclusions. In particular, although the lift of the best models are substantially similar, survival analysis modelling gives more valuable information, such as a whole predicted survival function, rather than a single predicted survival probability.

Furthermore, survival analysis modelling is dynamic in nature, and therefore, avoids biases that arise through time dependence, such as correlation of churn predictions with contract deadlines.

Finally, our results show that survival analysis modelling is a much powerful tool for lifetime value analysis and, consequently, for the actual planning of a range of marketing actions that impact on both perspective and actual customers.

We believe that survival analysis is a very promising tool in the area, and that further research is indeed needed, both in applied and methdological terms. From an applied viewpoint, directions to be further investigated concern the application of the methology to a wider range of companies (we have studies in progress in the banking sector). From a methodological viewpoint further research is needed on the robustification of Cox model which, being semiparametric and highly structured, may lead to variable results. We are investigating the usage of a Bayesian model averaging approach within this context.

To summarise, we believe that survival analysis modelling is a very promising tool in the field of customer lifetime values estimation, as it still is in medical survival analysis.

11. References

[1] Allison, P. D. *Survival Analysis Using the SAS Sysrem*, Cary, NC SAS Institute, (1995),.

[2] Anderson, K.M., *A non proportional hazards Weibull accelerated failure time regression model.* Biometrics 47, 281-288, (1991).

[3] Cox, D. R., *Regression Models and Life Tables*, Journal of the Royal Statistical Society, B34, 187-220, (1972).

[4] Giudici, P., *Applied Data Mining*, Wiley (2003).

[5] Hougaard, P., *Frailty models for survival data.* Lifetime Data Analysis. 1: 255-273, (1995).

[6] Kaplan, E. L. and Meier, R. *Nonparametric Estimation from Incomplete Observations*, Journal of the American Statistical Association, 53, 457-481 (1958).

[7] Klein, J.P. and Moeschberger, M.L., *Survival analysis: Techniques for censored and truncated data.* New York: Springer (1997).

[8] Keiding, N. Andersen, P.K., and Klein, J.P. *The role of frailty models and accelerated failure time models in describing heterogeneity due to omitted covariates.* Statistics in Medicine. 16: 215-224, (1997).

[9] Singer, Willet, *Applied Longitudinal Data Analysis,* Oxford University Press, (2003).

Neural Network to identify and prevent bad debt in Telephone Companies

Carlos André R. Pinheiro
Brasil Telecom-Coppe/UFRJ
andrep[ατ]brasiltelecom.com.br

Alexandre G. Evsukoff
Coppe/UFRJ
evsukoff[ατ]coc.ufrj.br

Nelson F. F. Ebecken
Coppe/UFRJ
nelson[ατ]ntt.ufrj.br

Abstract

This paper describes two main distinct results. The first is based on a cluster model to identify the insolvency behavior in a Telephone Company. Due to these clusters, the company can separate the customers into segments based on their characteristics and take different actions to increase revenue and avoid losses. Based on this knowledge about the customer, it is possible to monitor several groups of customers and perform distinct actions for each one. The second result is based on a set of predicting models to classify the insolvent customers. Using this model, the company can take preventing actions to avoid revenue leakage. A cluster model was based on an unsupervised learning, using Kohonen's self-organizing maps. Additionally, a MLP neural network was used for the predicting models. A distinct predicting model has been developed for each cluster identified by the first model, turning the input database more homogeneous, and increasing accuracy.

1. Introduction

Bad debt affects companies in all ways, generating impacts on operational processes, affecting the companies' investments and profitability. In the Telecom business, bad debt has specific and significant characteristics, especially in the Fix phone Companies. According to the regulation, even if the client becomes insolvent, the service should be continued for a specific period, which increases the losses [1][2].

The objective of this paper is to establish a bad debt preventing environment, creating models of patterns recognition related to the insolvent customers' behavior, as well as predicting models, with focus on the previous identification of non-payment events. This set of models aims to avoid revenue losses due to clients' insolvency.

2. Generating data sample

Brasil Telecom's data warehouse has been used to generate data base used in the construction of the models. In order to form and adjust variables stored in the data warehouse, aiming to develop mining models, it is necessary to implement an ETL process where, besides the data extraction, the transformation and loading of original information occur.

The first step for a data mining process is to have knowledge of the goals that should be attended and, from then on, identify the variables pertinent to these goals and, consequently, to the model to be developed. The variables associated to the development of models should be analyzed in order to verify the content of each information, with the execution of invariable and multivariable statistics, by defining typologies and choosing the roles of the variables in the models.

Brasil Telecom's base of clients has approximately 8 million clients, and almost 10 million of active terminals. However, according to the bad debt specific rules, only 5 million have been taken into consideration to develop the models presented here.

The first analysis is done in the data input interlace, where it is possible to establish the roles of the variables, their typology, size, format and label. The data input interlace presents a set of indicators for the interval variables and another one for the categorical variables. For the development of the neural network models, for the grouping as well as for the classification, a percentage of 5% and 10% of the real data have been used. The samples were extracted by using a random sampling algorithm. The Kohonen's self-organizing maps have been used for the grouping models, and subdivision patterns have been used in the receipt samples for the classification models, using artificial *multi layer perceptron* neural networks, being 40% of the data for the model training sample, 30% for the validation sample and 30% for the testing sample.

After the sample selection, data exploration and variable changing steps, the following step is the development of the models.

3. Self-Organizing Maps to cluster the insolvent customers

Since the insolvent customers' behavior is unknown, an unsupervised model will be used for the identification of the clients' most significant characteristics, which allows the creation of distinct groups based on the differences between them. The technique will be based on the self-organizing maps developed by Teuvo Kohonen in [5], which is widely used in unsupervised models, especially in those related to the identification of specific clusters.

The Kohonen map is created by using techniques based on neural networks. A set of vectors, which are also referred to as units, is repeatedly used as input for the map. Each unit is associated to a weight vector, initially consisting of random values. The units respond to the input vectors according to the correlation between the input vector and the unit's weight vector. The unit with the highest response is allowed to learning, as well as some other units in the "neighborhood". This neighborhood becomes smaller during the training period. The learning process occurs by adjusting the units' weights in small amounts, aiming to become more and more similar to the input vector. This process identifies an organization pattern in the map. Different units respond to different vectors, and the closest ones tend to respond to similar input vectors. When the training process is concluded, the set of input vectors is reapplied to the map, marking the unit of each input vector that responds more strongly to that input vector.

The first step is the creation of the data base, which has been made by means of a SAS code. This script creates a base in the Server and disposes the created information in a library, which can be shared by different models. The inclusion of the input interlace allows, besides creating meta-data, making changes in the variables roles, data type and format.

The second step is the creation of the data sampling to be used in the models construction and the sampling method to be considered. A simple sampling has been used for the proposed models, with 10% of the original base registries, meaning 493,285 clients. In the simple sampling process, it is possible that each observation of the data set be selected for the sample, independent on the other observations made. The stratification criteria of the sample are proportional, i.e., the proportion of observations in each stratification is the same in the sample and in the population as a whole. A frequency adjustment has also been chosen for the super-training. The frequency adjustment is applied when the stratification variable is responsible for the identification or target roles, which occurs with the classification models, where the target variable is to consider whether the client belongs to a "good" or a "bad" class.

The third step is related to the transformation of the variables available in the data sample generated. A set of derived variables has been generated with the original sample, using the SAS Code to create, for example, billing, bad debts, delays indicators and traffic usage behavior, or to create the arithmetic average of the same information described before.

However, several variables considered important for the model have an inadequate distribution to apply in neural network models or even self-organizing maps. For this reason, analyzing measures of skewness and kurtosis, it is possible to identify which will be the variables that will necessarily need some transformation. The measure of skewness gives the dispersion trend in the distribution of the variable values, i.e., an unbalance trend in the values symmetry. A positive skewness value indicates that the data located on the right of the average are more spread than the data on the left. A negative skewness value means the opposite. The measure of kurtosis indicates the disposition of the variable values distribution. A high kurtosis value means that data are very far from the average standard deviation. For this reason, when there are variables with high data dispersion, it is recommended to create transformed variables to diminish the asymmetry and normalize the data distribution. This normalization can be achieved by including the specific functions such as logarithm functions on the original variable or creating derived variables based on ranges or quarters of the original values.

Sometimes the creation of values ranges becomes extremely "adherent" to the established business rules for a specific mining model. Due to specific scenarios for the Telecom business, where the relation between the insolvency period and the interruption of services is not linear, it is important that the volume of clients allocated in the different ranges of delay period, allowing the definition of inhibiting actions and preventing events related to the revenue loss process. This helps establishing some insolvency period ranges, which will guide the decision taking in relation to the type of billing and the interruption of services. These ranges are, in fact, groups of 30, 60 and 90 days basically. Above this value, the client is allocated into another billing process, and can be considered as loss at some point of the time scale. However, aiming to create a wider view of the clients' behavior and the respective relation with the insolvency period, ranges with less interval have been created, gathering information for 15, 30, 45, 60, 75, 90, 120 and 150 days of delay.

In general, for indicators and average, values ranges or quarters have been created, establishing different content categories for each of them and, simultaneously, logarithm functions were applied on the original values, maximizing the normalization of the data.

Since the objective of this paper is to have a better understanding of the insolvent clients' behavior in a Telephone Company, i.e., identify the existing behavior pattern of the actions taken by the clients after becoming insolvent and, from then on, define behaviors categories according to the existing information in the data base, the data mining model chosen for this activity was the grouping model.

The fourth step is the modeling itself, where a set of different techniques can be tested aiming the best data distribution and, consequently, a grouping of clients with more similar characteristics. To create the grouping model using the self-organizing maps technique, a clustering model based on Euclidean distances has been developed. This method allows using criteria to identify an optimum number for the amount of possible groups. The automatic selection to identify the number of clusters can be used only for groupings based in square minimum criteria. The grouping method used in the implemented model was the *Centroid*, which establishes that the distance between two clusters is the squared Euclidean distance between the two centroids or means. The result of the clustering model was the identification of 5 specific groups. This grouping method defines the optimum number of groups to be identified by the cubic clustering criterion (CCC). This value was used to create the Kohonen model for self-organizing maps.

Based on the clustering model result, the configuration rules of the Kohonen model based on self-organizing maps were created. For the present model, an additional standardization method has been chosen, *standardize*, which specifies that the average is subtracted from the value of the variable, and that value is then divided by the standard deviation, multiplied by the dimension square root. Dimension here refers to the number of unique existing categories. For nominal variables, the default value of the dimension is C, which is the number of categories. For ordinal variables, the default value is 1.

There are some different ways of grouping, according to the function of the model. This function is related to the variable used to form the groups. It is default that the segmentation identifier is related to a function of the group. There are also possibilities of segmentation based on functions of models that use identification, input or target variables.

The method used for grouping was the Kohonen *Self-Organizing Map*, with a 5-line and 1-column topology. Generally, big maps are better but demand more processing time and effort in the training phase. The training method used was based on the learning rate. It is default that this percentage begins at 0,9 and is linearly reduced to 0,2 in the 1,000 first training steps. The learning rate should be between 0 and 1. With this method, it is also possible to define the maximum number of training steps (defined as 12,000 here), the maximum number of iteration (defined as 100), and the convergence criterion (defined as 0,0001). In this kind of procedures, one step is the processing of each case, i.e., in each line of information about the clients, the iteration is the processing of the whole data sample. Training is interrupted when any of the criteria defined previously is achieved. The selection criteria were based on centroids. The seeds initialization method was chosen as being the Main Component. In this method, the seeds are initiated in an equally spaced net in the plan of the two first main components. If the number of lines is minor or equal to the number of columns, the first main component is required to vary according to the number of lines. However, if the number of lines is bigger than the number of columns, the first main component is required to vary according to the number of lines and the second main component is required to vary according to the number of columns. The data input method in missing variables is the value of the closest seed, with missing values processing during the training phase based on averages for the interval, nominal and ordinal variables. In this method, the missing values are replaced by the average variable during the initialization of the grouping.

3.1. Clustering model results

There isn't necessarily a close relation among the number of groupings specified for the network architecture, the grouping mode parametering and the number of groupings identified in the data. For instance, by using a standard definition of 9 output unities, based on a square matrix of 3 lines and 3 columns, the neural network will allocate a set of 9 groups as output. However, during the neural network training, the model will be able to determine that there are only 3 groups of significant data. By analyzing the results of the groups, it would be possible to identify there 3 groups with a big percentage of the population, while the other groups would comprehend a small percentage of the whole population analyzed. This is a normal behavior for neural grouping, especially when referring to models based on distance. If it is necessary to obtain a more precise grouping, it is possible to create 16 outputs for the neural network. With that procedure, some of the registries that appeared in the same group in the first case would now be distributed among adjacent groups.

Since it is a method based on distances, with a maximum number of clusters previously defined, the first models can be executed based on a relatively large square matrix, with five lines and five columns or even more, totaling 25 clusters (maximum). By analyzing the variables of each group found, the clusters similarity is verified and an iterative work starts to search for the best configuration of the cluster matrix. With a continuous verification process, it is possible to reach an optimum number of groups with specific behaviors. Even if the best network configuration is a square matrix, an option would be to use rectangular matrixes, where the number of lines is different from the number of columns.

Another option, as described before, is to use segmentation models with criteria to identify the optimum number of the groups. This methodology can be applied to have a preliminary idea of the number of groupings that can be identified.

Applying the second option in the model adopted here, the optimum value was of 5 specific groups of insolvent clients. Here, the self-organizing maps matrix has 5 lines and 1 column, totaling five similar groupings.

In order to compose the model, i.e., the neural network, it is necessary to choose, given the available attributes of the data sample base, which will be the active variables, i.e., the ones that will be used in the network training process. It is also necessary to choose the variables that will be used to explain the results obtained when executing the model.

For this first segmentation mode, the variables described below have been used, in a definitive way, with some important results for the modeling. The variables used in the construction of the model were: branch, location, billing average, insolvency average, total days of delay, number of debts and traffic usage by type of call, in terms of minutes and amount of calls. Additionally, other variables have been created, such as the indicators and average billing, insolvency, number of products and services and traffic usage values. Derived variables were created, by means of maximum normalization, systematically using logarithm functions and by ranges of values in some cases. Then, some other relations were created, such as the insolvent clients' billing, among others generated, aiming to define the best set of information to describe the characteristics of the different groups of clients.

Therefore, for the scenario of the present work, the results below show how the data base of the company is grouped in relation to the profile of the insolvent clients. Analyzing the group, it is possible to have an extremely important knowledge of different behaviors in each client segment. This knowledge helps in the definition of collecting and revenue recovery actions that are more efficient than the execution of massive events.

The analysis of each group, as seen before, helps identifying common characteristics of a determined group of clients, allowing the development of specific politics for each of these segments, improving the performance of the company and the management of insolvent bills. However, the clusters analysis as a whole, comparing the variables of the model in the identified groups, allows a general view of the value of each group, which helps defining more adequate actions with the insolvent clients of the Company.

One of the most used comparisons in these cases is the relation between the billing and the insolvency values, that, related to the delay, provides a clear understanding of the clients' capacity of paying and, consequently, of the risk of insolvency. The relation among the distribution of the population and the billing, the insolvency values and the period of delay provides a capacity to recover revenue with actions focused on collection and, therefore, establish the return of the investment in data mining projects for the construction of models for insolvency prevention, risk of credit and payment possibilities.

Another important analysis is the existing relation between the insolvency average values and the average period of delay of each group or of each client classified in a determined group.

The first impression regarding the period of delay can lead to a conclusion that this situation can be positive for the Company, since the insolvent clients will be charged a fine. However, most part of the companies, especially the big ones, work with an extremely tight cash flow, with no gross margin or capacity financial resources. Therefore, for a reasonable period of delay associated with high insolvency values, it is necessary to obtain financial resources in the market. Since the cost of capital in the market is bigger than the fines charged for delay, the company has revenue losses, i.e., even if the company receives the fines, the cost of capital to cover the cash flow is higher.

Since the analysis are generally made based on average values, it is important to understand the distribution of some variables within the identified groups. For instance, the distribution of the period of delay can be analyzed taking into consideration the groups themselves, showing the average behavior of the clients within the group as well as in relation to the other clusters identified.

A second benefit, which is extremely important for the company, is the possibility to prevent insolvency events by previously identifying the characteristics of insolvent clients. The knowledge obtained from the neural grouping helps creating supervised learning models to predict the occurrence of insolvency, avoiding, therefore, a substantial revenue loss for the company. Therefore, two distinct areas of the company can be directly benefited by the data mining models developed: Collection, by creating different collecting politics, and Revenue Assurance, by creating insolvency prevention procedures.

Besides the implementation of neural grouping models aiming to create an effective information intelligence environment for the company, it is also necessary to develop a neural classification model. It provides the supervised capacity of learning the characteristics of the insolvent clients, their distinct behaviors and the steps taken by them until becoming insolvent. Then, it is possible to create the classification of these events and, consequently, avoid them, reducing the revenue loss for the company.

4. Neural network models to predict insolvency

The construction of the classification model, using an artificial neural network, basically follows the same steps described in the development of the grouping model. However, there is a significant difference, which is the inclusion of another step for executing the samples partitioning that will be used in the construction of the models. This partitioning is deeply related to the execution of the predicting models based on neural network techniques.

Therefore, after the input base step to the construction of the model and the inclusion of the creating interlace of the samples to be used, it is necessary to split the sample into three different data bases: training, validation and testing. The set of training data is used to train the neural network on the definition of the weights. The validating data set is used to manage the model and to assess the established estimates during the training process. Then, the testing data set is used to obtain an impartial and definitive value for the generalization error.

In the training mode, the function constructs a model based on a secondary set of selected input data. The remaining data is used to check the need for additional training for the developing model. The main objective of the model is to predict a value for the output field, the field related to prediction, specified for each registry in the data input. The training consists of alternating the learning and verification phases. The neural classification alternates these two phases to construct the most appropriate model.

The size of the sample represents the number of consecutive registries for selection from the input data to be used in the learning phase. While the network is learning, it calculates the weights used to represent the relations in the data. During the verification phase, the number of registries specified is highlighted. On the other hand, the size of the output sample represents the number of consecutive registries to be selected from the input data used in shifts training in the verification phase in order to determine if the desired accuracy objectives and the error limit have been affected. When these objectives are satisfied, the training mode is ended.

Precision is related to the accuracy rate, which is the percentage of right classifications. For the mentioned model, the rate used was 80%. When the network correctly classifies the percentage of specified registries in the input data, it obtains the precision objective. It calculates this percentage using the registries designed as output sample. If the error limit objective has also been achieved, the training ends and the research process stops. The network continues training until obtaining a wrong classification bigger than the specified percentage of registries of the testing sample.

The error rate is the maximum percentage of wrong classifications. Registries that could not be classified are not considered during the error rate calculation. It is also possible that the network has determined that the input fields represent an unknown class, which is not necessarily considered right or wrong. After reaching the specified errors rate, the training is ended and the research process stops, if the precision objective is also achieved.

In the testing mode, the function is applied to the same or new data with known class values to check if the trained network is providing correct results. In the application mode, the function uses a model created in the training mode to predict the specified field for each registry in the new input data. The set of input fields should be identical to the one used to generate the model [3][4].

The creation of a predicting model based on the neural network requires the generation of three different input bases: training, validation and testing. The partition interlace allows the creation of the three different data samples. The neural network model available in the *SAS Enterprise Miner* uses these samples in a distinct way. The training base is used for the preliminary adjustment of the model, where the analyst tries to find the best weights by means of this data set. The validation base is used to assess the adequacy of the mode, as well as for a fine adjustment.

This adjustment can be done not only in the construction of the models based on neural networks, but also in the construction of models based on decision and regression trees. For the neural network models, the validation base helps choosing the architecture or executing the training interruption algorithm. Then, the testing base is used to obtain a final and most impartial estimate for the model generalization error.

Two different models have been constructed based on neural networks. The selection criterion used in the first classification model, based on neural network, was the *Profit/Loss* method. This criterion chooses the model with the lower profit and loss average for the validation data set. The architecture chosen for the neural network was a multilayer perceptron one, where there are no direct connections between the neurons and the number of hidden layers depends on the data used for training. The neurons are optimized for data sets of high noise value. The number of preliminary executions helps finding good initial values for the weights of the neural network, and, in this case, the number of iterations was defined as 15 preliminary passages. The execution of a preliminary optimization helps avoiding the definition of initial weights that produce minimum places for the objective function. The training technique was chosen to be the *back-propagation* standard, and the processing time for the network convergence was defined as limitless, i.e., with no training time restrictions.

The neural network architecture includes the number of processing unities in the input and output layers, the number of processing unities in the used hidden layers and the number of processing unities in each of the hidden layers. The number of unities in the input and output layers depend on the input fields used as well as on the class field to be considered as target.

One or more processing unities in the input layer represent a numeric or categorical value of the input data. Each processing unity is connected to a processing unity in the following layer by a measured value that expresses the force of the relation. These measurements are called adaptable connections because their values are adjusted during the training to help the output network getting closer to the class values present in the data.

There are two ways of establishing how the neural network architecture will compose the model: automatically and manually.

In the automatic way, the research function evaluates distinct neural networks architecture with different numbers of hidden layers, including the processing unities of these layers. These alternative models are trained for a fixed number of passages and the best network architecture is selected for more training.

The automatic architecture demands more processing time that the manual one. The function constructs 5 different neural network architectures, with 0, 1 and 2 hidden layers. These network architectures are trained for a fixed number of passages over the data. The passages depend on the input size. The maximum number of passages is 50. After the training architecture passages are concluded, the network with better performance is selected for an additional training.

The manual architecture determination requires knowledge of the selection criteria for the neural network architecture, since the exact architecture should be specified in a very accurate way. The manual definition requires less processing time than for the automatic one, since only one neural network is trained by the function when the architecture is specified.

In order to determine manually which will be the neural network architecture, it is necessary to establish parameters for the learning and momentum rates. The learning rate controls the edge of the dynamical adjustment of the weights during the training process, aiming to improve the model convergence. Higher values indicate bigger changes, allowing faster network practices with higher values. However, the network accuracy can be lower in these cases. The momentum adjusts the change applied in one weight, resulting in updates of previous weights. This variable acts like a uniform parameter, which reduces the oscillation and helps keeping the convergence. Lower values can involve more training time and, consequently, more influence of the external data on weights.

In the second model, also based on neural networks and on advanced configurations, the topology of the network architecture is changed, but the mode selecting criterion is still based on gains and losses. In this new topology, the neural network architecture has two data input interlaces, allowing a better discretization of the ordinal and interval variables. The changed neural network architecture also has three hidden layers: the first has two interlaces, the second has three and the third has only one hidden interlace. Therefore, the error estimating process and the initial definition of weights in the input layer can be done in a more efficient way.

Like the previously defined network, the number of preliminary passages was of 15 iterations, and the backpropagation was chosen as the training methodology.

Additionally, this second neural network used an optimization step in the training process, where one preliminary optimization iteration is executed and then training is executed using the weights produced by the value of the best objective function of the preliminary optimization. The objective function was the maximum probability one, where the probability of negative logs in minimized.

However, the main difference between the two neural networks constructed were the results obtained. The second classification model based on artificial neural networks, with its changed architecture topology, has reached a significantly bigger hit ratio, especially concerning false positives and false negatives.

Using the neural network with changed topology in the second approach, the hit ratio in classifying good clients was 84%, and 81% in predicting bad clients. Therefore, the good clients' prediction error was 16% and the bad clients' prediction was 19%. Likely, it means that, from the really good clients, it was possible to predict that 84% were really good, presenting only 16% of errors. The model allowed predicting that 81% of the cases were really bad ones, with 19% of errors, classifying these clients as good. The average hit ratio was 82,5%, as shown in figure 53.

In the second approach using a changed neural network with a different architecture for the input and hidden layers, the graphics from the answerers, cumulative and non-cumulative, are presented below. The graphics from the answerers show that the neural network with changed topology has reached better results, with better answering rate. The non-cumulative analysis of the answerers based on the standard neural network model has obtained a positive answering rate in approximately 30% and 40% of the cases, while the non-cumulative analysis of the answerers based on the neural network with changed architecture has reached a positive answering rate around 40% and 50%, showing, therefore, a significant gain in positive answers.

Aiming to analyze the models constructed, especially the predicting ones, where the false-positive and false-negative are extremely important, it is possible to execute an assessment interlace in the SAS diagram models. The process of assessing the results is the last part of the data mining process, according to SEMMA methodology. The assessment interlace allows comparing the predicting models based on decision trees techniques, neural networks, regression, models grouped and defined by the users. The common assessment criterion is the comparison of gains and losses between the real values and the expected ones in the execution of the models. This type of criterion is highly intuitive and useful for a cross comparison as well as for making independent estimates of external factors, such as sample sizes, modeling interlaces and others.

The comparison statistics are computed during the training of the model and the best benefit obtained from the mutual assessment is the analysis of the results in terms of gains and losses, between the models and the answer baseline, taking into account random events. When it is necessary to develop a classification activity in data mining processes, the construction of several predicting models is widely adopted. In these cases, the use of the assessment interlace is even more important, especially when choosing the best cost/benefit model, or the best performance in relation to the investment made.

After assessing the results, the winning model (according to the gains and losses criteria), can be exported for the score interlace in order to be applied to new data bases.

The application of the winning model to a new data base consists of defining posterior probabilities for each level of target variables, whether binary, nominal or ordinal. For instance, suppose a binary target variable can only contain S and N values. In this case, the observation in the data set can be the definition of a posterior probability value of 0,8 for S and 0,2 for N. In this scenario, this observation can be typically classified as S. A second observation can have posterior probability values of 0,4 for S and 0,6 for N. In this case, this specific observation can be classified typically as N. The sum of posterior probabilities will always be 1 for each observation. The posterior probability values can be used for selecting the observation that is closer to a specific target. For instance, in the insolvency predicting model, with a binary target variable with G and B domains, (G are clients that will pay their bills next month and B are clients that will be insolvent next month), the posterior probabilities can be used for a more precise choice of clients with more chance to become insolvent. Therefore, for a collection or pre-billing action, the target public can be selected using the lowest posterior probability values, i.e., values closer to 0, which are related, in fact, to the B value.

5. Neural network models based on clustering

The models constructed until now bring great benefits for the company. The data mining process counted with the construction of three important neural network models, two based on Kohonen's self-organizing maps. From the models constructed, two were for segmentation and one for classification, and the combination of these models brings relevant financial benefits for the Corporation.

The first neural network model was constructed based on Kohonen's self-organizing maps and aimed to create a behavior segmentation of insolvent clients within the company. This model has identified five different groups with very distinct behaviors in relation to the use of telecom services, to the billing as well as to the assiduity in paying the bills. The groups identified varied from offenders to good clients, according to their payment characteristics, usage and period of delay in paying the bills. The segmentation mode proposed to identify bad debtors gives relevant knowledge on the insolvency behavior and allows the company to create specific collecting politics according to the characteristics of each group. The creation of different politics, according to the identified groups, allows generating a collection rule with distinct actions for specific periods of delay and according to the category of client. These politics can bring satisfactory results, not only concerning reducing expenses with collecting actions but also recovering revenue in a whole.

The second neural network mode developed was the classification of good clients and bad debtors, characterized as good and bad clients according to Brasil Telecom business rules. The rule used to classify clients as good or bad is described below. In case the client's period of delay is no longer than 29 days during the four months included in the observation window, i.e., October, November, December/2004 and January/2005 and if he also has positive value bills for the same period, he is then considered a good client. The classification model allows predicting who will become bad or good clients or, concerning the business rules presented, the clients that will be good or bad in the following month. The prediction of clients that will become bad debtors allows creating business processes to provide a significant expenses reduction for the company. Some of these procedures can be extremely simple but with high financial return. For instance, the suspension of billings issuance for clients that can highly become bad debtors inhibits the payment of taxes relative to those billings, which represent more than one third of the total value of the bill for Telephone Companies. Due to the high insolvency rate in the Telecom business, simply replacing the bill for billing warnings can save millions in financial charges. After the conclusion on the application of the models and the final analysis of the results, a financial viability study will be presented for the implementation of a corporative data mining corporative project, as well as a study on the expected investment return from the implementation of that project [7].

The third model developed was also a segmentation one but aiming to classify the clients according to characteristics such as services usage, payment behavior and generation of revenue. This classification can be understood as a score and, especially in this case, as a behavior score. The behavior score can be used in concession of credit politics, sales of services, offering service packs, acquisition and fidelization campaigns, among other applications. The behavior score can also be used to determine a value for the clients, especially in relation to their iteration with the company, not only in relation to sales actions, but also with general relationship and in client retention. As shown before, 10 score ranges have been defined according to their specific characteristics.

Therefore, the following step is to use the knowledge generated by the second segmentation model developed, the behavior score, to construct the predicting models using more homogeneous input bases, generating the expectation for more accurate results. We believe this will allow identifying more precisely which clients will become good ones, i.e., which clients will pay their bills and which will become bad ones, i.e., will not pay their bills.

5.1. Clustering to behavior segmentation

The segmentation model for scoring the clients has been developed similarly to the insolvent clients' identification model, with only one significant difference: the number of expected clusters. In the first model, a matrix with five lines and only one column has been defined, totaling five possible clusters to be identified. Also based on Kohonen's self-organizing maps, the second model matrix has five lines and two columns, totaling ten clusters to be possibly identified.

The predicting model to be constructed based on the segmentation applied to Brasil Telecom clients will allow creating neural network models that make use of more homogeneous training, validation and testing data sets, since the clients classified in a specific segment have similar characteristics and, consequently, variables containing domains and values closer from one another. This will help the input variable discretization process, the initial weights attributed to the input layers interlace and the estimative of retroactive error for the initial and intermediate layers. With that, we believe that the values of the layers interlaces can be more precise or closer to the real values.

Therefore, the base segmented in ten different categories of clients will be used as the original input data set for the construction of the neural models. To do so, it will be split into ten distinct input bases, which will be used by ten different classification models.

When using the score interlace in the behavior segmentation diagram, the base scoring process creates some additional variables. One of these variables is the segment identified for each client, i.e., the segmentation category applied to each registry in the input base. This variable is called _SEGMNT_, and will be used to stratify the data base into ten different input sets for the models construction.

This code segment will create ten different SAS bases that will be used as input data set in the construction of new classification models. The numbers of cases follows the quantitative sample generated by segmentation, i.e., with 51,173 clients in group 1, 22.051 in group 2, 17.116 in group 3, 23.276 in group 4, 17.178 in group 5, 9.902 in group 6, 23.827 in group 7, 48.068 in group 8, 19.194 in group 9, and finally, 14.868 in group 10, as shown below (table 7).

5.2. Predicting models based on stratified samples

Like the construction of the classification model, which has used a random sample of the whole base of clients, the ten predicting models, which use segmented samples of clients, will also have processing flows of information belonging to the assessed set of clients. However, for being distinct models, it is possible to adequate and specify adjustments for each model, such as topology configurations and network architecture, selecting criteria, error estimation techniques, among other parameters. This fact helps the specific adjustments to make the constructed predicting models more accurate and efficient, and the average hit ratio of the ten models can be higher than the unique model's hit ratio.

Therefore, using the samples created, ten different insolvency classification models have been constructed. The results are presented below.

The **first** classification model, which used the sample generated by cluster 1, generated the following error matrix. From the 51.173 observed cases, cluster 1 presented 50.943 observations about clients belonging to class **G** and only 230 observations about class **B** clients. The classification model constructed using the sample generated by grouping one had a hit ratio of 84,96% for **G**, i.e., predicted that 43.281 clients really belonged to class **G**, and 7.662 clients belonged to class **B**, resulting in a 15,04% of wrong predictions about good clients. Likely, the model had a hit ratio of 96,45% for class **B**, i.e., from the 230 observations of bad clients, the model predicted that 222 clients really belonged to class **B** and only 8 clients belonged to class **G**, i.e., with an error ratio of 3,55% about the bad clients observations. The **second** classification model, which used the sample generated by cluster 2, has generated the following error matrix. From the 22.051 cases observed, cluster 2 showed 21.800 observations of class **G** clients and only 251 observations of clients belonging to class **B**. The classification model constructed, which used the sample generated by grouping 2, had a hit ratio of 82,89% for class **G**, i.e., predicted that 18.070 clients really belonged to class **G**, and 3.730 belonged to class **B**, with an error ratio of 17,11%. Likely, the model had a hit ratio of 96,89% for class **B**, i.e., from the 251 bad clients observations, the model predicted that 244 clients really belonged to class **B**, and only 7 clients belonged to class **G**, i.e., with an error ratio of 3,11%. The **third** classification model, which used the sample generated by cluster 3, has generated the following error matrix. From the17.116 cases observed, cluster 3 presented 15.262 observations about the class **G** clients, and 1.854 observations about class **B** clients. The classification model constructed using the sample generated by grouping 3 had a hit ratio of 85,78% for class **G**, i.e., predicted that 13.092 clients really belonged to class **G**, and 2.170 clients belonged to class **B**, with an error ratio of 14,22%. Likely, the model had a hit ratio of 92,14% for class **B**, i.e., from the 1.854 bad clients observations, the model predicted that 1.708 clients really belonged to class **B**, and only 146 clients belonged to class **G**, i.e., with an error ratio of 7,86%. The **fourth** classification model, which used the sample generated by cluster 4, generated the following error matrix. From the 23.276 observed cases, cluster 4 presented 14.748 observations of class **G** clients, and 8.528 observations of class **B** clients. The classification model constructed, using the sample generated by grouping 4, had a hit ratio of 83,87% for class **G**, i.e., predicted that 12.369 clients really belonged to class **G**, and 2.379 clients belonged to class **B**, with an error ratio of 16,13%. The model had a hit ratio of 89,04% for class **B**, i.e., from the 8.528 bad clients observations, the model predicted that 7.594 clients really belonged to class **B**, and 934 clients belonged to class **G**, i.e., with an error ratio of 10,96%. The **fifth** classification model which used the sample generated by cluster 5, generated the following error matrix. From the 17.178 observed cases, cluster 5 presented 10.735 observations of class **G** clients, and 6.443 observations of class **B** clients. The classification model constructed, using the sample generated by grouping 5, had a hit ratio of 81,59% for class **G**, i.e., predicted that 8.759 clients really belonged to class **G**, and 1.976 clients belonged to class **B**, with an error ratio of 18,41%. Likely, the model had a hit ratio of 88,36% for class **B**, , i.e., from the 6.443 bad clients observations, the model predicted that 5.693 really belonged to class **B**, and 750 belonged to class B, with an error ratio of 11,64%. The **sixth** classification model, which used the sample generated by cluster 6, has generated the following error matrix. From the 9.902 observed cases, cluster 6 presented 6.119 **G** class clients observations, and 3.783 observations of class **B** clients. The classification model constructed, which used the sample generated by grouping 6, had a hit ratio of 80,92% for class **G**, i.e., predicted that 4.952 clients really belonged to class **G**, and 1.167 clients belonged to class **B**, with an error ratio of 19,08%. Likely, the model had a hit ratio of 88,57% for class **B**, i.e., from the 3.783 observations of bad clients, the model predicted that 3.351 really belonged to class **B**, and 432 clients belonged to class **B**, i.e., with an error ratio of 11,43%. The **seventh** classification model, which used the sample generated by cluster 7, has generated the following error matrix. From the 23.827 observed cases, cluster 7 presented15.816 observations of class **G** clients, and 8.011 observations of class **B** clients. The classification model constructed, using the sample generated by grouping 7, had a hit ratio of 82,09% for class **G**, i.e., predicted that 12.984 clients belonged to class **G**, and 2.833 clients belonged to class **B**, with an error ratio of 17,91%. Likely, the hit ratio was of 89,92% for class **B**, i.e., from the 8.011 observations about bad clients, the model predicted that 7204 clients really belonged to class **B**, and 807 clients belonged to class B, i.e., with an error ratio of 10,08%. The **eighth** classification model, which used the sample generated by cluster 8, generated the following error matrix. From the 48.068 observed cases, cluster 8 presented 47.707 observations of class **G** clients, and only 361 observations of class **B** clients. The classification model constructed, using the sample generated by grouping 8, had a hit ratio of 86,44% for class **G**, i.e., predicted that 41.238 clients really belonged to class **G**, and 6.469 clients belonged to class **B**, with an error ratio of 13,56%. Likely, the model had a hit ratio of 97,08% for class **B**, i.e., from the 361 observations about bad clients, the model predicted that 350 clients really belonged to class **B**, and only 11 clients belonged to class B, i.e., with an error rate of 2,92%. The **ninth** classification model which used the sample generated by cluster 9, generated the following error matrix. From the 19.184 observed cases, cluster 9 presented 15.967 observations of class **G** clients, and 3.217 observations of class **B** clients. The classification model constructed, using the sample generated by grouping 9, had a hit ratio of 88,69% for class **G**, i.e., predicted that 14.161 clients really belonged to class **G**, and 1.806 clients belonged to class **B**, with an error ratio of 11,31%. Likely, the model had a hit ratio of 91,32% for class **B**, i.e., from the 3.217 observations about bad clients, the model predicted that 2.938 clients really belonged to class **B**, and only 279 belonged to class **G**, i.e., with an error rate of 8,68%. The **tenth** classification model which used the sample generated by cluster 10, generated the following error matrix. From the 14.868 observed cases, cluster 10 presented 12.474 observations of class **G** clients, and 2.394 observations of class **B** clients. The classification model constructed, using the sample generated by grouping 10, had a hit ratio of 87,23% for class **G**, i.e., predicted that 10.881 clients really belonged to class **G**, and 1.593 clients belonged to class **B**, with an error ratio of 12,77%. Likely, the model had a hit ratio of 93,98% for class **B**, i.e., from the 2.394 observations about bad clients, the model predicted that 2.250 clients really belonged to class **B**, and 144 belonged to class B, i.e., with an error rate of 6,02%.

The final average hit ratio was of 84,45% of the cases, with an error ratio of 15,55% for class B, of the good clients. Likely, the average hit ratio for class **B**, of the bad clients, was of 92,38%, with an error rate of only 7,63%. Making a comparison with the classification model based on a unique data sample, the gain in the class G hit ratio was small, only 0,5%, however, the gain in the B class hit ratio was of 11,38%. Since the billing and collection actions are focused on class B clients, the gain in the good clients' classification ratio was not very significant, but the gain in the bad clients' classification ratio was extremely important, since it helps directing actions more precisely, with less risks of errors and, consequently, less possibility of revenue loss in different billing and collection politics.

Therefore, the classification model developed in the second approach, which was formed by ten different predicting models constructed based on stratified data samples by means of a behavior segmentation, had an average hit ratio of 84,45% for class **G**, i.e., of good clients, hitting 179.786 observations of good clients, considering the total sample used, making mistakes with 31.785 of the cases, and 92,38% for class **B**, i.e., of bad clients, hitting 31.552 of the observations and making mistakes with only 3.520.

As described before, the creation of ten distinct models for predicting insolvency has reached less than 1% gain in the classification of good clients. This fact may represent an innocuous effort in separating the data bases according to the behavior of different groups of clients, and the later use of these data bases in the construction of separate models. However, the gain in the classification of bad clients was higher than 11 in relation to the unique model, which represents a significant improvement in predicting bad clients accurately, i.e., in predicting insolvency cases. Given the total volume of the data base of a Telecom Company, with a high number of clients, this gain may represent millions (Reais) in revenue recovery or avoiding financial expenses in billing and collecting actions.

Combining the results of the behavior segmentation model, which has split clients into ten different groups according to the subscribers' characteristics and the use of telecom services, with the classification models that predict the class of the clients G (good) or B (bad), we observe that each of the identified groups has a very peculiar distribution of clients between classes G and B, which is, in fact, another group identification characteristic.

6. Predicting models comparison

As seen before, the first classification model was constructed based on an input data set that contemplates all Brasil Telecom clients. This model has reached an average hit result of 87,5%, with 94% average hitting for good clients and 81% for bad ones. In the second methodology, ten distinct classification models have been developed, each of them having a stratified sample as a subset of input data, based on the behavior segmentation executed for all clients. The average hit ratio of this models was 84,45% for the good clients class and 92,38% for the bad clients.

This result confirms that the use of more homogeneous samples empowers the data discretization, improving, therefore, the definition of the initial weights and consequently the errors estimation. With a better choice of the initial weights and a more accurate errors estimation process, it is possible to improve the learning capacity of the supervised model based on a neural network. These improvements make the model more especial, without losing, however, the capacity of prediction generalization when it is necessary to apply it to other data base. The set of characteristics, specification and sensitivity indicates how trustworthy the predicting model is concerning accuracy and application to different data observation. The verification of these capacities can be checked in a graphic known as ROC curve. It shows how the classification model behaves in relation to the matches and errors of the predicted classes versus the real observation. This behavior can be observed with the errors matrixes presented for the models previously constructed and analyzed.

In the classification cases, where the target variable is binary, the error matrix presents the number of predicted cases as belonging to class 0 and that really belong to class 0, called true negative, the number of predicted cases as belonging to class 0, but belong in fact to class 1, called false negative, the number of predicted cases as belonging to class 1 and that really belong to class 1, called true positive, and the predicted cases as belonging to class 1, but belong in fact to class 0, called false positive.

An important analysis in comparing the performance of the classification models is the graphic that represents the prediction gains and losses. This graphic shows the average profit in the peak of the first decile, which can be presented in a cumulative or non-cumulative way for each percentile. The graphic below shows that the accumulated gains of the composed neural network model have values higher than the unique neural network model, and both have gains when compared to the base line. In models with great predicting capacity, the profits will reach the top in the first decile. However, observing the values of accumulated profit in relation to the base line, we observe that the unique model profit is the same up to percentile 40, while the composed model keeps the profit for the whole sample, i.e., up to percentile 100. This means that the second model can bring return even if applied for the whole population, and that the first must be applied to only 40% of the base of clients.

7. Action plans to revenue assurance

One of the most important points in data mining processes and finding out knowledge is the use of advanced results. Besides the steps of the process as a whole, such as the identification of the business objectives and the functional requirements, the extraction and transformation of the original data, the data manipulation, mining modeling and the analysis of the results, there is a very important step for the Corporation, which is the effective implementation of the results in terms of business actions.

This action plan, which is executed based on the knowledge acquired from the results of the data mining models, is responsible for bringing the real expected financial return when implementing projects of this nature.

Several business areas can be benefited by the actions that can be established. Marketing can establish better relationship channels with the clients and create products that better meet their needs. The Financial Department can develop different politics for cash flow and budget according to the behavior characteristics of the clients concerning payment of bills. Collection can establish revenue assurance actions, involving different collecting plans as well as credit analysis politics.

The use of the results of the models constructed, even the insolvent clients' behavior segmentation or the subscribers' summarized value scoring, as well as the classification and insolvency prediction can bring significant financial benefits for the company. Some simple actions defined based on the knowledge acquired from the results of the models can help recovering revenue for the Corporation, as well as avoid debt evasion due to matters related to regulations.

The first model, the insolvent clients' behavior segmentation, allows creating specific collecting actions and helps recovering revenue related to non payment events. This cluster model has identified five groups of characteristics, some of them allowing distinguish collecting actions, avoiding additional expenses with collecting actions.

Nowadays, the collecting actions occur according to temporal events, according to the defined insolvency rule. Generally, when the clients are insolvent for 15 days, they receive a collection notice. After 30 days, they receive a billing notice and part of the services are interrupted (they are not able to make calls). After 60 days, they receive new billing notices and the services are completely interrupted (they cannot make and receive any calls). After 90 days, the clients' names are forwarded to a collecting company which keeps a percentage of the revenue recovered. After 120 days, the insolvent clients' cases are sent to judicial collection. In these cases, besides the percentage given to the collecting company, part of the revenue recovered is used to pay lawyers fees. After 180 days, the clients are definitely considered as revenue loss and are not accounted as possible revenue for the company.

8. Financial gains

Each action associated to this collection rule represents some type of operational cost. For instance, the printing cost of the blocking letters sent after 15 days of delay is approximately R$ 0,10. The partial and total blocking letters printing cost R$ 0,05. The average posting cost for each letter is R$ 0,70. The negativation processes cost approximately R$ 0,80, considering the existing *bureaus* in the market. When the clients are delayed for more than 110 days, they are forwarded to another collecting company. Besides the costs mentioned above, around 10% and 20% of the revenue recovered is paid to these companies, depending on the delay range. After 180 days of delay, the clients are considered as losses and are forwarded to judicial collection, where the discounts for paying the bills vary from 30% and 80%, also depending on the delay range of the client.

Analyzing the groups of insolvent clients identified by the segmentation model, we observe that there are classes of subscribers whose behavior is not offensive to the company, especially when this behavior is related to the payments of the collecting rule used by the Company. For instance, considering the percentage of insolvent clients with 30 days of delay, i.e., 27,61% of approximately 5 million clients, some representative numbers can be found. Group 1 has 74.091 insolvent clients with delay until the first billing notice is sent. Based on this behavior, it is possible to assume that these clients would pay their bills with delay independent on receiving the notice. This group has an average of 3 unpaid bills, totaling 222.273 issuing. Group 2 has 57.543 insolvent clients within the notification period, with an average of 3 unpaid bills, totaling 172.692 issuing. Group 2 has 98.197 insolvent clients within the notification period, with an average of 3 unpaid bills, totaling 294.591 issuing. Group 4 has 64.012 insolvent clients within the notification period, with an average of 2 unpaid bills, totaling 128.024 issuing. Group 5 has 46.647 insolvent clients within the notification period, with an average of 3 unpaid bills, totaling 139.941 issuing.

Therefore, among the insolvent clients, 340.490 would pay the bills with some delay without any actions from the Company, which means that 957.458 notices would not be issued and posted. Taking into account that the average printing and posting cost is around R$ 0,80, the financial expenses monthly avoided only by inhibiting the billing notice issuing would be of approximately R$ 765.966,00, assuming that these clients make the payments with delay even before the collection and partial blocking notices are issued.

The same procedure could be applied to clients that pay their bills within 30 or 60 days of delay, i.e., before the partial and total blocking occur. However, it is extremely difficult to estimate the interference or the effect of the 15 days notices when the bills are paid with fifteen and thirty days of delay. It is also difficult to evaluate the effect, not only of the fifteen days notice but also of the partial blocking when bills are paid with thirty and sixty days of delay. However, the possible savings related to expenses with collecting actions for bills paid within thirty and sixty days of delay would not be accounted in the revenue recovered with the implementation of the data mining models presented in this paper.

However, there are some actions related to classifications and predicting models that can avoid significant expenses for the Company, especially those related to taxes paid when issuing the bills. For instance, the Telephone Companies are obliged to pay value-added taxes on Sales and Services when issuing the telephone bills. Therefore, independent on the client paying the bill or not, when the bills are issued, 25% of the total amount is paid in taxes.

Therefore, an action inhibiting issuing bills for clients that highly tend to be insolvent can avoid a considerable revenue loss for the Company, since it will not be necessary to pay the taxes. There are basically two types of financial evasion. The first refers to the values that were not received for the use of the telephone network, i.e., relative to the calls made within the Company's concession area or when using the CSP (Communications Services Provider). For instance, a client makes a local call or a LDN using Brasil Telecom's CSP 14, and does not pay the telephone bill. Consequently, the company does not receive the payment for the services rendered. The second – and most offending - type of revenue evasion refers to the values paid for interconnection with other companies, when the calls are made between different companies. This occurs when a Telephone Company uses another Telephone Company network. For instance, if another Company subscriber calls a Brasil Telecom subscriber, the part of the calls made within Brasil Telecom network must be paid by the originating Telephone Company to Brasil Telecom. Therefore, if a Telemar subscriber calls a Brasil Telecom subscriber, the part made within Brasil Telecom's concession area must be paid by Telemar to Brasil Telecom.

Since the issuing of the bill implies on the tax payment, the above mentioned action avoids revenue loss.

In this case, the insolvency classification model can be very useful, as shown below. Consider the clients' segment with prediction values for class G between 0 and 0,1, which, inversely, are the clients that tend more to move to the B class, i.e., with a hit ratio between 90 and 100%. Also consider the precision level hit ratio of 81% for the bad class observations, predicted as really belonging to the bad class. This ratio was obtained from the unique insolvency classification model developed before.

The insolvency probability ranges show how close the clients are to becoming insolvent for more than 29 days or not, i.e., how close they are from classes G or B. However, in a general way, the predicting model classified the clients base with a specific distribution, according to the business rules related to those classes. Figure 69 shows a population distribution achieved by the predicting model according to the business rules related to the definition of classes G or B. The model classified 72,21% of the clients as being good ones and 27,29% of bad ones.

The combination of the information mentioned before, i.e., the clients trend to become insolvent or not according to the established probability ranges, the behavior characteristics of each group identified and the population distribution according to G and B classes brings new knowledge of the business, which can be used to define new corporate collecting politics. Figure 70 shows the Good and Bad classes distribution for each of the behavior groups identified.

Each clients segment, i.e., each client with predicting value in the ranges of values between 0 and 0,1, 0,1 and 0,2, 0,2 and 0,3, 0,3 and 0,4, 0,4 and 0,5, 0,5 and 0,6, 0,6 and 0,7, 0,7 and 0,8, 0,8 and 0,9, and 0,9 and 1, has a set of common characteristics, as if they were real clusters of clients. One of these characteristics is the monthly average billing for each range. Thus, each range of client identified by the predicting model has an average billing, as described in table 13.

Considering the population distribution of each range of client identified by the predicting model, and extrapolating the population percentages identified for the whole clients data base considered in the construction of the models for the present paper, we observe the following amounts of individuals in each score range, as shown in table 14.

Taking into account only the range with more insolvency probability, i.e., with score between 0 and 0,1, there are 852.891 clients. Since the monthly average billing of these clients is of R$ 113,00, the total possible value to be identified and related to insolvency events is R$ 96.376.700,00. Since the hit ratio for the Bad class observations was 81% in the unique classification model, the total clients whose telephone bills issuing would be inhibited for the first segment was 690.842 cases. Based on the same billing average, the value relative to non-payment events is R$ 78.065.127,00. Consequently, the total value that can be avoided by inhibiting the telephone bill issuing and not paying the issuing taxes, considering only the bills with more probability for non-payment, which means more than 90%, is R$ 19.516282,00. Taking into account a linear and constant behavior of the insolvent clients base, which has been happening , in fact, for the last years, the value that can be avoided for the company in terms of revenue is monthly.

Naturally, there is a risk in not issuing a bill for a client that would make the payment, due to the classification error inherent to the classification model. However, it is necessary to emphasize that the financial gain, even considering the range of clients with a non-payment probability of more than 90%, and an error ratio of 19% of the classification model, it would be significantly compensating. For instance, in this scenario, the monthly balance between avoided loss and revenue not billed would be around R$ 5.782.602,00, i.e., even if the bill is not issued for clients that could make the payment, the avoided loss compensates a lot the non-effective revenue. However, this risk can be minimized by defining different business actions concerning the results achieved by the developed data mining models. For instance, by issuing collection notices instead of bills for the clients with specific score ranges or by including only the clients with 99% probability of no payment. With this action, the risk of not issuing a bill for a client that could possibly pay the bill (even if remote) is minimized. Obviously, this is a business decision related to revenue gains and losses in the insolvency events of the company.

For instance, taking into consideration the best classification range, i.e., the range of clients with more probability for insolvency, with a hit ratio between 95 and 96%, the percentage of the population of this range of clients consists in 9,75% of the observations, i.e., 480.954 cases. For this range, the average billing was approximately R$ 97,00. Therefore, the possible revenue recovery would be R$ 46.652.505, 00, and applying the classification error of the model, this population range falls to 389.572 cases, diminishing the possible return to R$ 37.788.529,00. The amount due to taxes is 25%, i.e., R$ 9.447.132,00. Taking into account the classification error of the model, the population wrongly predicted consists of 91.381 clients, with a loss of R$ 6.647.982. The final result of the financial gain achieved with the inhibiting issuing actions for this range of clients would be R$ 2.799.150,00.

Therefore, taking into consideration the worst case, the revenue recovery would be around R$ 2.799.150,00 due to inhibiting the bills issuing and not paying the taxes, plus R$ 765.966,00 due to inhibiting the 15-days billing notices printing and posting. The first return results from the classification models and the second one results from the segmentation models. Therefore, the monthly financial return expected with the implementation of the proposed data mining models is around R$ 3.565.116,00.

By using the second methodology, constructing ten different classification models, the error ratio in classifying bad clients is of 7,62% of the cases, inhibiting a higher number of issuing for clients that will not make the payments. Therefore, for the same classification range, i.e., for the clients with a probability between 95 and 96% for insolvency and considering that the population percentage of this range is 9,75%, the total cases to be considered is 480.954 clients. For this range, the average billing is R$ 97,00. The possible revenue recovery are the same R$ 46.652.505,00. However, applying the classification model error, which is much lower in the second methodology, this population range falls to 444.305 cases only, diminishing the expectation for return to R$ 43.097.584,00. The tax rate due is 25%, i.e., R$ 10.774.396,00. Taking into consideration the model classification error, the wrongly predicted population is 36.649 clients, with a billing loss of R$ 2.666.191. Thus, the final financial gain achieved with the inhibiting actions for this range of clients would be of R$ 8.108.205,00.

The difference between the two methodologies concerning revenue recovery, i.e., the predicting model based on a unique input sample, contemplating all clients of the base, and a set of predicting models, each of them related to a group with distinct behavior characteristics, meant gains of approximately R$ 5.365.038,00.

It is necessary to emphasize that these are monthly gains over the involved capital, and not the total value effectively, since the Company can recover part of this amount with the income tax return by the end of the fiscal year. The gain is, in fact, on the cost of capital acquisition in the market, necessary to cover cash flow deficits occurred due to values not received from insolvent clients.

9. Conclusion

The development of segmentation models helps acquiring some knowledge about the clients' behavior according to specific characteristics, which can be grouped, allowing the company to define specific relationship actions for each of the identified groups, associating the distinct approaches with some specific characteristics. The analysis of the different groups allows identifying the distribution of the main characteristics of each of them according to business perspectives, allowing the Company to develop market intelligence. This intelligence, which is extremely analytical, helps defining clearer and more objective business rules, allowing the Corporation to achieve better results with less cost and efforts.

The classification models allow the Company, based on the knowledge obtained from identified grouping, to anticipate specific events, becoming more pro-active and, consequently, more efficient in the business processes.

In this paper, the insolvent clients' behavior segmentation model Brasil Telecom to have more specific knowledge related to the insolvency scenario of the Company. This knowledge helps defining more focused actions against insolvency, as well as creating more efficient collecting procedures. The insolvency behavior segmentation model has identified five characteristic groups of clients, which were separated in distinct classes. The *moderate*, with an average usage of services behavior and average commitment to paying the bills; the *very bad*, with an average usage of services behavior, but low commitment to paying the bills; the *bad*, with a high usage of services and relative commitment to paying the bills; and the *offender*, with high usage of services behavior and average commitment to paying the bills.

The segmentation model of all Brasil Telecom clients helped identifying the value of each client for the company and allowed defining more efficient relationship actions. Each characteristic group has a peculiar behavior concerning the usage of services and products of the Company, with distinct events of telephone bills payment. The ten groups of clients identified can be segmented according to the value they aggregate to the Company, allowing the creation of more specific products, more focused relationship actions and more objective billing and collecting activities management.

The classification models, even the unique predicting model or the predicting models based on the input sample segmentation, allow the creation of a highly benefic business intelligence for the Company, enabling the Corporation to create pro-active actions to identify and prevent non payment of bills, and define a more efficient billing and collecting actions management. The classification models helped identifying the average percentage of good and bad clients according to a defined business rule and, therefore, monitoring these percentages on a long term basis.

9. References

[1] ABBOTT, Dean W., Matkovsky, I. Philip, Elder IV, John F., *An Evaluation of High-end Data Mining Tools for Fraud Detection*, IEEE International Conference on Systems, Man, and Cybernetics, San Diego, CA, October, pp. 12-14, 1998.

[2] BURGE, P., Shawe-Taylor, J., Cooke, C., Moreau, Y., Preneel, B., Stoermann, C., *Fraud Detection and Management in Mobile Telecommunications Networks*, In Proceedings of the European Conference on Security and Detection ECOS 97, London, 1997.

[3] FAYYAD, U. E., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., *Advances in Knowledge Discovery and Data Mining*, AAAI Press, 1996.

[4] HAYKIN, Simon, *Neural Networks – A Comprehensive Foundation*, Macmillian College Publishing Company, 1994.

[5] KOHONEN, T., *Self-Organization and Associative Memory*, Springer-Verlag, Berlin, 3rd Edition, 1989.

[6] LIPPMANN, R.P., *An introduction to computing with neural nets*, IEEE Computer Society, v. 3, pp. 4-22, 1987.

[7] PINHEIRO, Carlos A. R., Ebecken, Nelson F. F., Evsukoff, Alexandre G., *Identifying Insolvency Profile in a Telephony Operator Database*, Data Mining 2003, pp. 351-366, Rio de Janeiro, Brazil, 2003.

[8] RITTER, Helge, Martinetz, Thomas, Schulten, Klaus, *Neural Computation and Self-Organizing Maps*, Addison-Wesley New York, 1992.

Data Mining-Based Segmentation for Targeting: A Telecommunications Example

Kasindra Maharaj and Robert Ceurvorst
Synovate, Decision Systems
222 Riverside Plaza, Chicago, IL. 60606
Kasindra.Maharaj[ατ]synovate.com

Abstract

Companies have become increasingly focused on the return on investment (ROI) for every endeavor. In order to boost ROI of product development and marketing initiatives, they must develop a more integrated understanding of consumer behavior, needs, attitudes and demographics and then leverage that understanding to target their initiatives more efficiently and profitably. That need has fueled strong growth of data mining advances and applications, which in turn have allowed companies to deal with increasingly complex problems.

This paper illustrates using an example data set how a data mining process we call Targeted Segmentation can help companies in any industry achieve that integrated understanding of consumers. It does this by marrying information of various types and possibly from various sources in a manner that ensures that the resulting segments are logically and strongly differentiated on all the types of information in the analysis. The key is not "throwing variables into an analysis" but rather exploiting the relationships that exist between information of different types.

Our case study is from the Telecommunications industry, where a large service provider sought need- and benefit-driven segments that could be identified with at least 90% accuracy using only behavioral and demographic characteristics on their database. For purposes of comparison, we produced behavioral and attitudinal segmentations, in addition to the targeted segmentation. Targeted and behavioral segmentations yielded comparable degrees of differentiation (both much stronger than attitudinal segments), but only targeted segments were predicted well -- with better than 90% accuracy -- using the client data alone. Further, with targeted segments, 49% of the customers generated 70% of the company's profit. With behavioral segments, 43% of customers yielded only 58% of profit and with attitudinal segments, 40% of customers produced only 44% of profit. Targeted Segmentation has produced similar types of results in numerous industries, including financial services, healthcare, automotive, fast-moving consumer goods, and others.

Published by
Department of Mathematics and Computing Science

Saint Mary's
University

Halifax, Nova Scotia, Canada

Technical Report Number: 2003-02 October, 2003
ISBN: 0-9734039-0-X