Machine Learning, 57, 83–113, 2004

Lessons and Challenges from Mining Retail E-Commerce Data

RON KOHAVI* ronnyk@cs.stanford.edu
Amazon.com, 1200 12th Ave. South, Suite 1200, Seattle, WA 98144

LLEW MASON lmason@bluemartini.com
RAJESH PAREKH rparekh@bluemartini.com
Blue Martini Software, 2600 Campus Drive, San Mateo, CA 94403

ZIJIAN ZHENG* zijianz@microsoft.com
Microsoft Corporation, One Microsoft Way, Redmond, WA 98052

Editors: Nada Lavrač, Hiroshi Motoda, Tom Fawcett

Abstract. The architecture of Blue Martini Software's e-commerce suite has supported data collection, data transformation, and data mining since its inception. With clickstreams being collected at the application-server layer, high-level events being logged, and data automatically transformed into a data warehouse using meta-data, common problems plaguing data mining using weblogs (e.g., sessionization and conflating multi-sourced data) were obviated, thus allowing us to concentrate on actual data mining goals. The paper briefly reviews the architecture and discusses many lessons learned over the last four years and the challenges that still need to be addressed. The lessons and challenges are presented across two dimensions: business-level vs. technical, and throughout the data mining lifecycle stages of data collection, data warehouse construction, business intelligence, and deployment. The lessons and challenges are also widely applicable to data mining domains outside retail e-commerce.

Keywords: data mining, data analysis, business intelligence, web analytics, web mining, OLAP, visualization, reporting, data transformations, retail, e-commerce, Simpson's paradox, sessionization, bot detection, clickstreams, application server, web logs, data cleansing, hierarchical attributes, business reporting, data warehousing

1. Introduction

E-commerce is an important domain for data mining (Kohavi & Provost, 2001), with massive amounts of clickstream and transactional data that dwarf in size data warehouses from a few years ago (Kimball & Merz, 2000).

At Blue Martini Software, we had the opportunity to develop a data mining system for business users and data analysts from the ground up, including data collection, creation of a data warehouse, transformations, and associated business intelligence systems that include

*The author was previously at Blue Martini Software.

reporting, visualization, and data mining. The system was made available to clients in 1999 and has since been purchased for e-commerce by brand-name retailers, such as Bluefly, Canadian Tire, Debenhams, Harley Davidson, Gymboree, Kohl's, Mountain Equipment Co-op, Saks Fifth Avenue, Sainsbury, Sprint, and The Men's Wearhouse.

Focusing on the retail e-commerce domain allowed us to provide solutions to some tough problems. For example, Pfahringer (2002) wrote that one of his lessons from participating in the KDD Cup (an annual data mining competition) was that "Every problem is different! There is no such thing as a standard problem." Kohavi (1998) suggested that one way to cross the chasm from academia to commercial data mining was to build a vertical solution and complete the chain around data mining from collection to cleaning, mining, acting, and verifying. The e-commerce and data mining architecture (Ansari et al., 2001) we built provided us with unique capabilities to collect more data than is usually available for data mining projects. From the very beginning, significant consideration was given to data transformations and analysis needs. This can be contrasted with one of the challenges facing business intelligence in situations where analysis is performed as an afterthought. In these cases, there is often a gap between the potential value of analytics and the actual value achieved because limited relevant data were collected or because data must go through complex transformations before they can be effectively mined (Kohavi, Rothleder, & Simoudis, 2002). We limit further discussion of the architecture to areas where the additional information provides context for sharing the lessons learned. More details about the architecture are available in Ansari et al. (2001).

The lessons described in this paper are based on data mining projects we completed during the past four years. Over this time we have analyzed data from more than twenty clients. The durations of each of these projects varied from a few person-weeks to several person-months. Results of some of these projects are available as case studies or white papers such as MEC Case Study (Blue Martini Software, 2003a), Debenhams Case Study (Blue Martini Software, 2003b), and eMetrics Study (Mason et al., 2001). Although all of the client data analyzed can be classified broadly as retail e-commerce, the clients' businesses were often significantly different from each other, coming from multiple industry verticals, with varying business models, and based in different geographic locations (including the US, Europe, Asia, and Africa). Sources of data included some or all of customer registration and demographic information, web clickstreams, response to direct-mail and email campaigns, and orders placed through a website, call center, or in-store POS (Point-Of-Sale) systems. Depending on the client, the quantity of data analyzed varied from a few thousand records to more than 100 million records, and the time period of the data varied from a few months to several years. Despite these differences, all the lessons we describe are general, in the sense that they are valid across many of the clients we have analyzed. In fact, we believe that many of the lessons we describe are also widely applicable to data mining domains outside of retail e-commerce.

The paper is organized as follows. Section 2 begins with high-level business lessons and challenges. Section 3 describes technical lessons and challenges on data definition, collection, and preparation. Section 4 presents technical lessons and challenges for analysis, including experimentation, deployment, and measurement. We conclude with a summary in Section 5.

2. Business lessons

Our goal in designing the software was to make it easy for an organization to utilize business intelligence capabilities, including reporting, visualizations, and data mining. We consider the data mining lifecycle stages in the following natural order: requirements gathering, data collection, data warehouse construction, business intelligence, and deployment (closing the loop). In each of the following subsections we describe our approach, the lessons learned and the challenges that merit further investigation. It should be noted that this section includes lessons and challenges from the perspective of the business user or the broader organizational unit that is the main sponsor of the data mining projects. Lessons that deal with the more technical aspects will be described in Sections 3 and 4.

Our clients, i.e., the "businesses," expect a seamless integration of business intelligence capabilities with the software for the channels we provide, namely the website, call center, and campaign management, while allowing relatively easy integration with their other sources of data.

2.1. Requirements gathering

The process of gathering the requirements for data analysis is critical to the eventual success of any data mining project. Since all the clients we focused on are from the retail e-commerce domain, we have developed significant experience in this domain and a clear understanding of the business terminology. We learned the following lessons related to the requirements gathering phase.

1. *Clients are often reluctant to list specific business questions*. In quite a few of our engagements our clients did not give us any specific business questions. Sometimes, they do not even know what questions to ask because they do not understand the underlying technology. Even when we specifically asked them to give us questions, they simply asked us to find some interesting insights. The importance of involving the business users has been previously documented (Berry & Linoff, 2000). Our lesson here is the value of whetting the clients' appetite by presenting preliminary findings. After an interim meeting with basic findings, the clients often came up with quite a long list of business questions they wanted us to answer.
2. *Push clients to ask characterization and strategic questions*. Even when the clients did present us with business questions, they were basic reporting type questions. Clients had to be pushed to formulate deeper analytic questions. For example, the question asked initially would be something like "What is the distribution of males and females among people who spend more than $500?" or "What is the response rate of the last email campaign in each region?" instead of asking "What characterizes people who spend more than $500?" or "What distinguishes the people who responded to the last email campaign from those who did not?" This lesson is aligned with the CRISP-DM (Chapman et al., 2000) recommendations on business understanding and with Berry and Linoff (2000) who write, "Defining the business problem is the trickiest part of successful data mining because it is exclusively a communication problem."

It is worth mentioning that formulating questions through interacting with business experts is part of the data mining process. While providing example questions to business people and educating them on data mining could be useful in many cases, developing some methodology and best practices to help them define appropriate questions remains challenging.

2.2. Data collection

Data collection at a website includes clickstreams (both page views and session information), customer registration attributes, and order transactions (both order lines and order headers). From a business perspective, the collection in the Blue Martini architecture is mostly transparent. Page views are tied (through database foreign keys) to sessions, which are determined by the application-server logic, obviating the need for sessionization (see Section 3). Transactions are automatically tied to sessions and to the customers. All the data are automatically recorded directly into the database, avoiding the need to collect web logs from multiple web servers, parse them, and load them. These are all possible because the architecture is "aware" of higher abstraction levels, unlike stateless http requests seen by web servers.

Many of the data preparation steps described by Cooley, Mobasher, and Srivastava (1999), which put significant burden on organizations that would like to mine their data, are unnecessary when using this collection architecture (or a similar architecture that collects data at the application-server layer). One of the reasons why we have seen so much research around web logs and sessionization is that the web logs were designed to debug web servers and not necessarily to provide useful data for data mining.

The architecture also records the following unique "business events":

1. Every search and the number of results returned. This allows us to produce often requested reports on searches that return too many results and hence need dedicated results pages and "failed searches" (zero results returned) that help improve the search thesaurus and merchandising (e.g., they help identify early trends) for products the merchandisers are not aware of.
2. Shopping cart events (add to cart, change quantity, and delete). These are very hard to discern from web logs, yet are automatically handled by the architecture. The availability of these events makes it easy to track shopping cart abandonment.
3. Important events such as registration, initiation of checkout, and order confirmation. These provided data for computing micro-conversion metrics (Lee et al., 2001).
4. Any form field failure. The architecture supports a validation regular expression for every form field. Validation failures are recorded to help with usability testing. One example of the usefulness of this event happened with one of our clients. Two weeks after deployment of our system, we looked at form failures and found thousands of failures on the home page every day! There was only one form field on the home page: a place to enter your email address to join their mailing list. Thousands of visitors were typing search keywords into the box and because these search keywords failed to validate as email addresses, the architecture logged them. To fix the problem, the client simply added a clearly identified search box on the home page and set the default contents of the email box to the word "email."

The important observation is that the collection of these events is transparent. While website designers may not think about collecting form field failures as they concentrate on the aesthetics, the architecture automatically collects them because they are useful for analysis.

The architecture also collects additional attributes that are not commonly available, such as the user's local time zone, whether their browsers accept *gzip'ed* content (useful for robot detection), color depth, and screen resolution. Screen resolution and color depth help identify more technical users who are using advanced hardware and are potentially heavier spenders. These attributes were useful for customer segmentation. We have learned two important lessons with respect to data collection and management.

1. *Collect the right data, up front.* Changes to operational systems typically pass through multiple business processes, and the time taken for requested changes to actually appear in production is often lengthy. This means that the time taken to get data collection right the first time is typically dwarfed by the time taken to make changes later. In our framework, we often collect data that have no use within the transactional system itself, and are collected exclusively for analytical purposes.
2. *Integrate external events.* There are many external events that fall outside the realm of data collection per se, but can have a large impact on data analysis. For example, in the retail domain, marketing events like promotions or advertisements are often not directly captured within the transactional system, and are thus not found within the collected data. However, these marketing events can have dramatic effects in terms of patterns within the data. False conclusions can easily be drawn if these external events are not taken into account. In the KDD-Cup 2000 competition (Kohavi et al., 2000), we provided the Gazelle.com marketing calendar to contestants as many of the marketing events were correlated with patterns in the data. Yet several of the participants reached incorrect conclusions because they neglected to look at the marketing calendar.

In summary, the architecture has served us well and has solved many practical issues that would otherwise make data mining of e-commerce data extremely difficult.

2.3. Creation of the data warehouse

The creation of a data warehouse requires significant data transformations from an operational system, sometimes called On-Line Transaction Processing, or OLTP (Kimball, 1996). It is often quoted that 80% of the time to complete an analysis is spent in data transformation (Piatetsky-Shapiro et al., 1996). In our application, the analytics component and production component including the website and call center are well integrated and we control the data sources in the production subsystem, allowing us to automate the creation of the data warehouse for those channels.

A process that we call *DSSGen* generates the Decision Support System database automatically, based on the meta-data that we have available and specific operations, such as denormalizations and pivots that we coded.

Our clients have been able to use DSSGen successfully and get the transfer to work in a matter of days, compared to several months of work for common Extract-Transform-Load (ETL) tools. Custom changes to the production site are automatically reflected in

the generated data warehouse thereby dramatically reducing the maintenance costs. For example, if in a client implementation, ten new customer attributes are added based on the client's unique registration form, these will automatically be transferred and made available in the data warehouse.

From a business intelligence perspective, the process of automatically creating the data warehouse was extremely successful. Here are some of the challenges we face:

1. *Firewalls*. Firewall issues continue to complicate the implementation because the website is usually in a demilitarized zone (DMZ) (Cheswick & Bellovin, 1994), which has restricted access. This means that on secure implementations that are now common, copying of files across firewalls must be customized.
2. *Integration*. Integration with other data sources still requires the "standard" significant effort. There is little we can do here, except to point to available ETL packages. This may have been an easier route had we used a standard ETL package for our own transfers, but we believe that, in such a case, we could not have provided the tight integration we provide today with DSSGen.

2.4. Business intelligence

Business Intelligence includes reporting, visualizations, and data mining. In our architecture, we provided clients with an industry standard report writer (Crystal Reports), visualizations, and data mining algorithms that included rules induction, anomaly detection, entropy-based targeted statistics, and association rules. Since different activities require data transformations, we also developed a very powerful transformation engine with an accompanying graphical user interface (GUI).

On the positive side, we were able to derive very interesting insights from clients such as Mountain Equipment Co-op (Blue Martini Software, 2003a) and Debenhams (Blue Martini Software, 2003b). For example, we found that merchandisers were far from optimal in assigning cross-sells and that product associations using association rules (Agrawal & Srikant, 1994) were much better; we identified characteristics of customers who were low spenders but were likely to migrate to a higher tier of heavy spenders; we showed that for Mountain Equipment Co-op, flat-fee shipping was superior to free shipping for revenues and profits, at least in the short term, etc.

We also learned several important lessons worth sharing:

1. *Expect the operational channels to be higher priority than decision support.* Insight will come later. When businesses were building their websites and call centers, they were typically backlogged with things that were urgent for these operational channels and that left little time for analysis. Sites went live, call centers were taking calls, but business intelligence was going to be done once things were more stable, often months later.
2. *Crawl, walk, run*. The most immediate need for our clients was basic reporting, not fancy analytics. The businesses were trying to understand basic metrics related to their website performance and needed more out-of-the-box reports such as dashboards of key performance indicators and summary reports, which we started to provide more and

more with each release. We found that providing out-of-the-box reports is one way to jump-start the business intelligence process.

3. *Train data analysts.* There is clear recognition now that a large database requires a good Database Administrator (DBA). However, data mining has a "magical" aura surrounding it. Unrealistic expectations about "press the button and insight will flow out" need to be reset. For people to do data mining effectively they need to be properly trained and this takes time and effort. Data mining methodologies like CRISP-DM (Chapman et al., 2000) can help here.
4. *Tell people the time, not how to build clocks*. In "Built to Last" (Collins & Porras, 1994), the authors suggested that building clocks to tell time is far more important than telling people in time. We found the opposite to be true—clients wanted interesting insights, and early on did not care how we found them. Because many of the insights we discovered generalized well across multiple clients, it was easy to show a graph that depicts how online spending correlates with distance from their physical stores (the farther you live from the nearest retailer's physical store, the more money you spend on the average purchase) than to explain how we found it. Over time we started to develop standard reports that are available out-of-the-box. These reports include interesting findings and highlight insights that make a difference to the business.
5. *Define the terminology.* Our clients often ask questions such as: What is the difference between a visit and a session? How do you define a customer? Did every customer purchase? Why does there appear to be a difference between the same metrics in different reports? Are orders from our Quality Assurance (QA) department included in the revenue, even though the shipping address for all these orders is specified as 555 Foobar Ave and we know not to ship to this address? Writing a good glossary and sharing the terms across reports was something we learned the hard way.

While we learned from the above lessons and believe they can be reasonably addressed, several significant challenges remain elusive:

1. *Make it easier to map business questions to data transformations*. Mapping business questions to data transformations is a complex task today. Can we make that easier? While we built a user interface that supports many useful transformations, fundamental operations like aggregations remain a complex concept to grasp. When System-R was initially conceived at IBM San Jose Research Lab (now Almaden Research Center) in the early 70s, they were designing a language (now SQL) for non-programmers. In a System-R reunion Don Chamberlin (1995) said "What we thought we were doing was making it possible for non-programmers to interact with databases." The SQL92 standard is now 600 pages and the SQL99 standard is 1,000 pages. Is it possible to build a transformation language and a user interface that is significantly easier to learn?
2. *Automate feature construction*. Feature construction requires a mix of domain knowledge and a data miner's expertise. While we are able to provide many features out-of-the-box for our domain, with every client we build hundreds of unique attributes as a customer signature against which to run. These include features specific to their site design, product mix, etc. Can these be easier to construct automatically?

3. *Build comprehensible models*. The goal of data mining is to provide business users with interesting insights. We have restricted ourselves to building models that are easily understood, such as decision trees, decision rules, and Naïve-Bayes. Are there other models that one can build, which are easy to understand by business users?
4. *Experiment because correlation does not imply causality*. When interpreting data mining results it is often the case that correlation is confused with causality. Business users need to be made aware that correlation does not necessarily imply causality. For example, when analyzing the benefits of online search functionality that is provided on our clients' sites, it is immediately apparent that visits with search on the site have a higher average session length than those that do not. However, upon carefully examining the data, one can see that in order to perform a search on the site a session must have at least two page views, one to type in the search string, and the other to view the search results that are returned (if any). So to truly compare whether people who search on the site spend more time than those who do not, we must first filter out all visits of length one which can account for 50% of all visits. It turns out that even after excluding visits of length one there is still a strong correlation. To establish a causal relationship, one should conduct control/treatment experiments.
5. *Explain counter-intuitive insights*. On a few occasions it becomes difficult to present insights that are seemingly counter-intuitive. For instance, when analyzing a client's data we came across an example of Simpson's paradox (Simpson, 1951). Simpson's paradox occurs when the correlation between two variables is reversed when a third variable is controlled. We were comparing customers and looking at their channel preferences, i.e., where they purchased. Do people who shop from the web-channel only spend more on average as compared to people who shop from more than one channel, such as the web and physical retail stores? The line chart in figure 1 shows that for each group of shoppers who shopped once, twice, three times, four times, five times, and more than five times respectively, the average spending per customer on the web-only channel is more

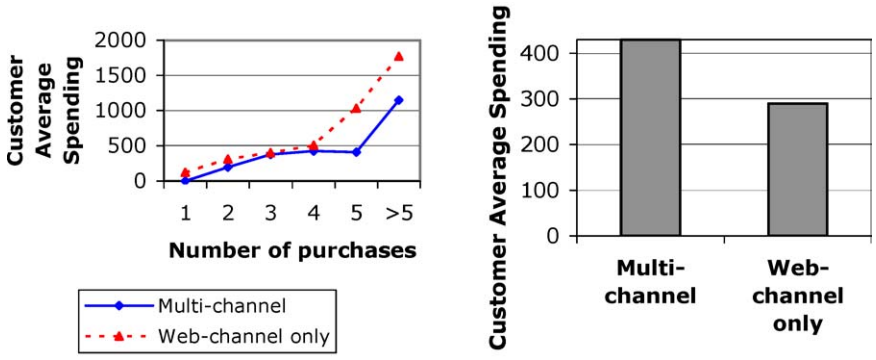


Figure 1. Average yearly spending per customer for multi-channel and web-only purchasers by number of purchases (left), and average yearly spending per customer for multi-channel and web-only purchasers (right). Web-only customers dominate multi-channel customers in their spending in *all* segments showing number of purchases (left), yet they spend less on average (Simpson's Paradox).

than the average spending per customer on multiple channels. However, the bar chart in figure 1 shows that the average spending per customer for multi-channel customers is more than that of the web-only channel. This reversal in the trend is happening because a weighted average is being computed and the number of customers who shopped more than five times on the web is much smaller than the number of customers who shopped more than five times across multiple channels. Such insights are often difficult to explain to business users.

6. *Assess the ROI of insights.* It is difficult to assign a quantitative value to determine the return on investment (ROI) of the insights that are obtained from data mining. In some cases, the insights are directly actionable in which case one can measure the impact of taking the recommended action. For example, in the case of one large automotive manufacturer, they managed to measure the effect of changes to their website that were suggested by our analysis. These changes directly resulted in a 30% improvement in revenue. However, in other cases, the insights might be related to improved browsing experience or better customer satisfaction, the results of which are hard to measure quantitatively. It could make things even harder when they have different or opposite short term effects and long term effects.

2.5. Deployment and closing the loop

Insight is only useful if it is shared across the organization and utilized. There are two ways to utilize insights and models:

1. *Share insights*. Insights obtained by data mining should be shared across the organization. Our initial products required people to install client software. In later releases we converted to a browser-based "Analysis Center," where reports and data mining results can be viewed and shared. Users do not need to install anything, just enter a URL into the browser, which made the reports and analyses much more accessible.
2. *Take action*. Score visitors by their likelihood of response, implement a product recommender (e.g., based on associations), and improve the interface on the site based on events. All these things can be done if people see the value. Many of these things are not done often because it is hard to automate the deployment of results. Our experience showed that it is useful to have the architecture providing easy ways to implement product recommendations based on associations and to score customers based on models.

One of the challenges that we see in this area is:

- *Have transformed data available for scoring*. Scoring customers on something of interest implies that information needs to be collected at the touch point (point of interaction with the customer), data transferred to the warehouse, customers scored, and scores transferred back to the operational touch points to close the loop. This cycle with off-line analysis is good for coarse decisions, and developed models are useful for some types of real time choices such as showing different products and images when users return to the website

later on. However it is not appropriate for dynamic actions (e.g., recommendation based on the purchase a few minutes ago). Conversely, building a model requires significant data transformations and only simple models can be built without access to transformed data. It is usually too expensive and complex to transform the data at the operational side. Is there a middle ground that is useful? Some companies, such as E. piphany (http://www.epiphany.com), provide real-time scoring and learning based on very simple models.

3. Data definition, collection, and preparation

In Sections 3 and 4 we drill into the technical details and discuss the low-level lessons learned from data and analysis related issues. Business intelligence efforts must have clear metrics to evaluate success. Once these goals and metrics are defined, organizations must strive to collect the appropriate data, clean and transform them, and make them available for analysis. We divide our discussion of data-related lessons and challenges into three sub-sections: data collection and management, data cleansing, and data processing.

3.1. Data collection and management

For any organization, data collection should be driven by its business intelligence goals. As mentioned in Section 2, data collection is often an afterthought (Kohavi, Rothleder, & Simoudis, 2002) and this restricts the type and depth of analysis that can be performed.

1. *Collect data at the right abstraction levels*. Most web analytics are performed using web logs, collected by the web server. The web logs were generated mainly for the purpose of debugging the web server. As a result, they are "information-poor" and also require significant pre-processing before they are a useful data source for analysis. Web servers are stateless, each page requested is served independently, but all of the pages viewed in a single visit to the website are logically grouped together in a session. Much effort has been expended on devising reliable methods for this process of "sessionizing" web logs (Spiliopoulou et al., 2003). We completely bypassed the issue by having the application server (where the application logic is executed) log the clickstreams and sessions, rather than the web server. The architecture uses cookies to track sessions, and if cookies are not available, it rewrites the URL for each hyperlink to keep track of the session for the visitor (ensuring that the next click belongs to the same session).
2. *Design forms with data mining in mind.* Significant time and effort is spent in designing forms that are aesthetically pleasing. The eventual use of the collected form data for the purpose of data mining must also be kept in mind when designing forms. Analysis of Gazelle.com data (Kohavi et al., 2000) revealed a very large number of female customers. Even customers with male names had their gender recorded as female. The registration form defaulted the gender field to "female" and many customers did not bother to change the default value. To collect unbiased data, the form design must not specify any default value and ask customers to select a value.

3. *Validate forms to ease data cleansing and analysis.* In the electronic world, form data is a big source of data errors. Appropriate form validation can save a lot of time needed for data cleansing and later data analysis. Some example data types that can be validated automatically include date, time, phone numbers, postal addresses, and age (check the value range). For domain data types, use drop down lists instead of free text fields. For example, use a drop down list containing "Decline", "Male", and "Female" for gender.
4. *Determine thresholds based on careful data analysis*. Session timeout duration is an important threshold for clickstream collection. It determines the duration of inactivity after which a session would be considered timed out. In prior work that characterized browsing strategies for users on the World-Wide Web it was determined that the mean time between two browser events across all users was 9.3 minutes and a session timeout threshold of 25.5 minutes ($1\frac{1}{2}$ standard deviations from the mean) was recommended (Catledge & Pitkow, 1995). Analysis of clickstream data from a large clients' website revealed that several user sessions were experiencing a timeout as a result of a low timeout threshold. These users got an unpleasant timeout message and lost their active shopping cart. (More recent versions of the software save the shopping cart automatically at timeout and restore it when the visitor returns.) However, even if it were not for the loss of the shopping cart, identifying a session for analysis purposes is very important because breaking the session could result in an item added to the shopping cart and the checkout process being assigned to different sessions, even if the user experience was that of a single (long) session. Figure 2 shows the impact of different session timeout thresholds set at 10-minute intervals on two large clients. For this experiment we designate a visitor session as having timed out prematurely if the visitor does not make a page request for a duration longer than the timeout threshold, yet comes back (makes a request) in less than 3 hours. If the session timeout threshold were set to 25 minutes then for client A (left chart in figure 2), 7.25% of all sessions would experience timeout and 8.6% of sessions with active shopping carts would lose their carts as a result. However, for client B (right chart in figure 2), the numbers are 3.5% and 5.1% respectively. Thus,

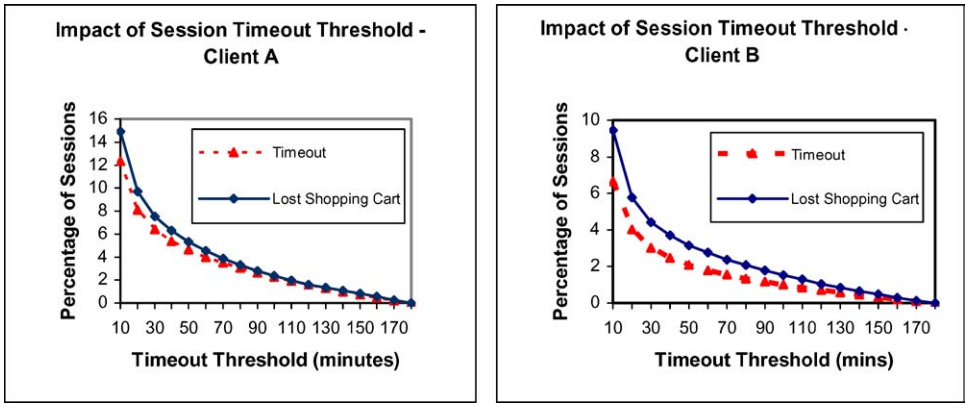


Figure 2. Setting a suitable session timeout threshold.

clients must determine the timeout threshold only after careful analysis of their own data. Further, the smooth curves in both the charts do not suggest any threshold beyond which the impact of session timeout would be minimal for either of the two clients. As a general rule, we recommend that the session timeout for e-commerce sites be set to no less than 60 minutes. That is more than double the sessionization time recommended in Catledge and Pitkow (1995). When sessionizing for analysis purposes (as opposed to operational concerns about keeping sessions in memory), the referrer field in the request can be used to further improve the process. If the referring page is another page on the site being analyzed, the allowed threshold could be large (e.g., 120 minutes) because the user has clearly clicked on a link in the current site, while a request with an external referrer could be used to initiate a new session when the gap is longer than 60 minutes.

The following challenges apply to data collection and management:

1. *Sample at collection*. Large e-commerce websites generate on the order of 10–100 million page views in a single day. Logging data for every single request and every session is very expensive in these cases both in terms of the load on the system that logs the data to the database and the space requirements for storing so much data. An obvious question to ask is whether or not to sample at the source (Domingos, 2002). We provided our clients with the ability to sample clickstream collection. Although sampling would effectively address the two issues mentioned above, it introduces new problems. Sampled data will not be able to accurately capture *rare* events such as searching for a particular term or credit card authorization failure. Further, business requirements, such as payment for advertising clickthrough referrals, require that exact (rather than approximate) statistics are available. Can we provide enough flexibility to apply sampling intelligently, while still capturing rare events or required statistics with full accuracy?
2. *Support* "*slowly changing dimensions.*" Visitors' demographics change: people get married, their children grow, their salaries change, etc. With these changes, the visitors' needs, which are being modeled, change. Product attributes change: new choices (e.g., colors) may be available, packaging material or design change, and even quality may improve or degrade. These attributes that change over time are often referred to as "slowly changing dimensions" (Kimball, 1996). The challenge is to keep track of these changes and provide support for such changes in the analyses.
3. *Perform data warehouse updates effectively.* Can we manage efficient and timely updates to the data warehouse without interrupting availability of results to business users? Further, what are good guidelines for deciding how much data to retain in the data warehouse and when to purge older data?

3.2. Data cleansing

Data cleansing is a crucial prerequisite to any form of data analysis (Fayyad et al., 1996; English, 1999). Even when most (or all) of the data are collected electronically, as in the case of e-commerce, there can be serious data quality issues. Typical sources for these data quality issues include software bugs, customization mistakes, or plain oversights in (any or

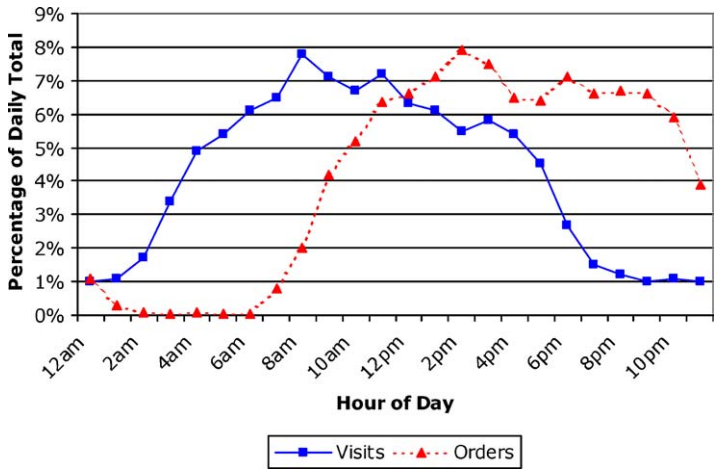


Figure 3. Distribution of visits and orders by hour-of-day. The 5-hour lag was traced to time-zone problems.

all of) the software, implementation, system configuration, data collection, or data transfer (Extract-Transform-Load, or ETL) process.

In cleansing data collected by our clients, we have learned a valuable lesson:

- *Audit the data.* We've found serious data quality issues in data warehouses that should contain clean data, especially when the data were collected from multiple channels, archaic point-of-sale systems, and old mainframes. An example of this is shown in figure 3 that shows the orders and visits by hour-of-day for a real website. Orders seem to "follow" visits by five hours, whereas we would expect visits and orders to be close to each other in time. It turned out different servers were being used to log clickstream (visits) and transactions (orders), and these servers' system clocks were off by five hours. One was set to GMT and the other to EST.

There are several significant challenges related to data cleansing in the e-commerce domain:

1. *Detect bots.* Web robots, spiders, and crawlers, collectively called bots, are automated programs that visit websites (Heaton, 2002). Typical bots include web search engines (like Google), site monitoring software (like Keynote), shopping comparators (like mySimon), email harvesters (like Power Email Harvester), offline browsers, and computer science students' experiments. Due to the volume and type of traffic that they generate, bots can dramatically change clickstream patterns at a website, in turn skewing any clickstream statistics. For example, on several of our client websites we observed that the average page views per visit when bot visits were excluded was 1.5 to 2 times the average page views per visit when bot visits were included. It must be pointed out that in the clickstream collection described here bots appear in short, mostly single request sessions instead of a single long session that can span several days. Even on high volume retail e-commerce sites, between 5% and 40% of visits are due to bots. Identifying bots

in order to filter out their clickstreams is a difficult task since they often do not identify themselves adequately or pretend to be real visitors, and different bots can generate radically different traffic patterns. Current bot filtering is mostly based on a combination of a continuously tuned set of heuristics and manual labeling. It is worth mentioning that *page tagging* methods of clickstream collection (Madsen, 2002), which execute blocks of javascript at the client's browser and log the statistics returned by the javascript at a server, avoid bots because they require the execution of javascript, which bots rarely execute. However, people who do not have javascript turned on in their browsers or who click on a link before the javascript code can download and execute will not have their visits correctly logged by page tagging systems. These visits can amount to about 5% of all human visits, thereby resulting in inaccurate clickstream statistics.

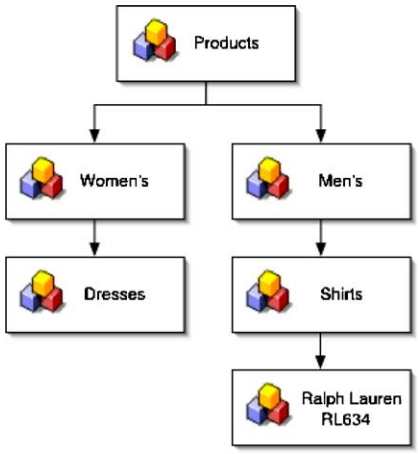
2. *Perform regular de-duping of customers and accounts.* Transactional systems usually do not provide safeguards to stop the generation of duplicate customer records. Some businesses also have the notion of accounts, where the mapping between customers and accounts is a many-to-many relationship (one customer may have multiple accounts, or one account may be shared amongst multiple customers). Additional difficulties in identifying unique customers arise in e-commerce systems from the availability of kiosks that are used by multiple people to log on to the website. The fact that customer records might not have enough information to reliably distinguish unique customers poses significant challenges in reliably merging or "de-duping" customer records.

3.3. Data processing

Most analytical algorithms, and software packages that implement these algorithms, provide limited support for many of the complex data types that occur commonly in retail e-commerce data. Examples of "complex" data types include attributes that are hierarchical, cyclical, or include multiple notions of "missing" values. Specialized data processing is often required to effectively mine this type of data. We will first discuss some lessons in dealing with these types of attributes, and then list some challenges.

1. *Support hierarchical attributes*. In retail, products are commonly organized into hierarchies in a product catalog: SKUs (Stock Keeping Units) represent the most fine-grained definition of a product (e.g., individual SKUs for different colors of the same product), and are derived from products, which are derived from product families, which are in turn derived from product categories. An example product hierarchy is shown in figure 4. Hierarchies of SKUs, products, families, and categories are typically between three and eight levels deep, and contain between three and ten items at each level—meaning that the total number of unique SKUs could range from a few hundred to several million. A customer purchases SKU level items, but generalizations of purchasing behavior are likely to be found at higher levels (e.g., families or categories). Another example of a hierarchical attribute that occurs frequently in retail data is geography, where the hierarchy could be zip code, city, state, and country.

   For learning algorithms that do not directly support hierarchical attributes, a product hierarchy could be exposed to the algorithms using attributes representing the product



#	Women's	Men's	Shirts	Dresses
1	T	F	F	F
2	F	T	T	F

Figure 4. An example product hierarchy (left) and a pivot for a "Women's" item and a "Men's Shirt" item (right).

family and category. These attributes will often contain many distinct values. Some approaches have been proposed to deal with set-valued attributes (Cohen, 1996) and for improving the efficiency of algorithms that deal with hierarchical attributes (Aronis & Provost, 1997). An alternative, which works well in practice, is to perform what we call a hierarchy pivot. For each node in the product hierarchy that was identified as interesting by the business user, we create a boolean attribute in the purchased line item. The value of the boolean attribute indicates whether the SKU belongs to that node in the hierarchy. Figure 4 shows two example records, the first representing a SKU under the "Women's" node (but not under "Women's Dresses"), and the second representing a SKU under the "Men's Shirts" node.

2. *Handle cyclical attributes*. Due to the transactional nature of retail e-commerce systems, date and time attributes occur frequently (e.g., account creation date, order placement date, and web visit date). However, the common date/time format containing the year, month, day, hour, minute, and second is rarely supported directly by data mining algorithms. Even when it is supported, a date/time attribute is typically treated as a single continuous variable, which rules out the discovery of interesting date and time patterns that generalize to predicting future events. Patterns that generalize most often involve the time *delta* between two dates (e.g., order placement and shipment date), or are based on the *hierarchical* or *cyclical* nature of dates and times (e.g., hour-of-day). In order to effectively mine date and time attributes, data transformations are required to compute time intervals between dates, and to create new attributes taking into account the hierarchical and cyclical nature of dates and times. We found that transforming date-time attributes to multiple attributes, such as hour-of-day, day-of-week, week, day-of-month, month, and quarter is useful in supporting the discovery and visualization of cyclical patterns, such as "Saturday traffic is high." It is worth noting that businesses often define their own date cycles based on marketing or financial processes,

meaning that using standard calendar date cycles to build attributes may fail to uncover patterns.
3. *Support rich data transformations*. We found that it is necessary for a data analytics package to provide integrated transformations with a suitable user interface. Examples of transformations are aggregation, row filtering, column deletion, new column creation with an expression builder, and binning which is the process of mapping integers and real-values to discrete values. Working with these kinds of transformations saved us significant time in data processing.

Some challenges related to processing data include:

1. *Support hierarchical attributes.* Supporting hierarchical attributes is important in practice (see bullet point 1 above). A few algorithms have been designed to support hierarchical attributes directly (Almuallim, Akiba, & Kaneda, 1995; Aronis & Provost, 1997; Zhang, Silvescu, & Honavar, 2002), but they do not scale to large hierarchies. The process of automating the (now manual) process of utilizing hierarchies effectively still remains challenging.
2. *Handle* "*unknown*" and "*not applicable*" *attribute values.* The assignment of attributes, such as size, weight, or color to products is common in e-commerce, and they are often used for restricting search results ("show me all extra-large shirts"), or the grouping of products for display based on common attribute values. These product attributes are coincidently also valuable for data mining, since generalizations can be found based on the attributes of products, rather than on just the particular type of product. However, some attributes may apply to some classes of products, but not to others. For example, size makes sense for clothes and shoes, but not for books. For books, the size attribute would have a NULL value. In this case, NULL means "not applicable", rather than "unknown", and needs to be treated differently (Quinlan, 1989). In fact, NULLs in database have multiple interpretations and semantics. The ANSI/SPARC interim report (ANSI/X3/SPARC, 1975) lists 14 of them. We supported this distinction between our two interpretations for NULLs using meta-data. For every attribute, meta-data would determine whether a NULL value should be treated as either "not applicable" or "unknown." "Not applicable" is a distinct value for mining purposes, whereas "unknown" implies that the attribute is relevant but its value is unknown. For example, if a site adds a registration question that asks new customers registering for their gender, then all customers who registered prior to this addition should have a NULL value that implies "unknown." Our solution, while providing a step forward, does not address cases where an attribute may need both an "unknown" and a "not applicable." For example, a "mega-pixel" attribute is applicable only to digital cameras, hence a NULL should imply not applicable; however, for some digital cameras the value may be unknown because the manufacturer did not specify it. The need to have two types of NULLs is complicated by the fact that databases support only one NULL. For real-valued columns in a database, one must resort to special values (e.g., negative infinity) to denote the second semantic NULL, causing problems in aggregate functions like sums and averages. Very few data mining algorithms, with the notable exception of C5.0 (RuleQuest Research, 2003), can correctly accommodate this subtle difference.

4. Analysis

This section presents the technical lessons and challenges based on our experience with analysis of data from our clients. We organize these lessons by the following phases of the data analysis process:

1. understanding and enriching the data,
2. building models and identifying insights,
3. deploying models, acting upon the insights, and closing the loop, and
4. empowering business users to conduct their own analyses.

4.1. Understanding and enriching data

The first step after getting the business questions from the client is to get a good overview of the data. We cannot emphasize how much value we have derived from just getting a feel for the data and getting to know what tables are available, what attributes belong to each table, what the different attributes mean, and how they relate to each other.

1. *Statistics*. Elementary statistics including the distribution of each attribute, the number of NULL and non-NULL values, the minimum, maximum, and mean value for each continuous valued attribute are useful in obtaining an overview of the data and for identifying anomalies. Further, in cases where we plan to build predictive models to predict a discrete target such as *campaign responder* or *heavy spender*, it is beneficial to run targeted statistics that give an idea about the degree of correlation between each attribute and the target. We found it extremely useful to order the columns by their information gain (an entropy-based metric) (Quinlan, 1986), highlighting the most critical columns first. Figure 5 shows a lift chart for the target that indicates whether there was a search in the visit in relation to *screen resolution*. Overall, 10.5% of all visits searched (i.e., had

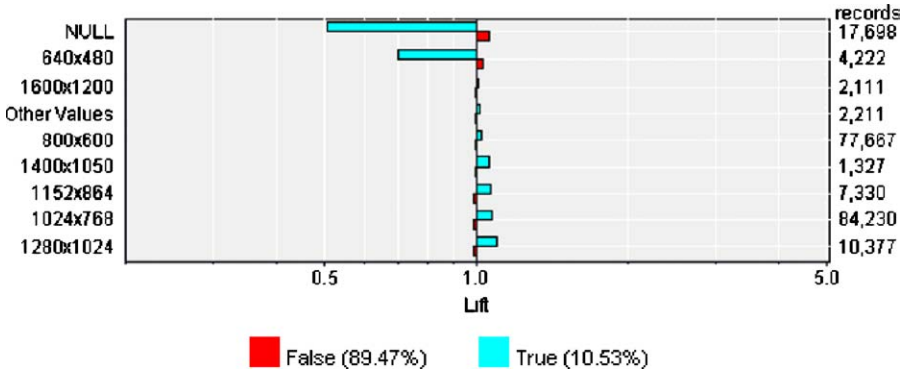


Figure 5. Lift chart showing relationship between screen resolution and search. NULL means the browser returned no information.

the target value true). The chart shows lift greater than one for the commonly used screen resolutions such as 1280×1024, 1024×768, and 800×600 implying that visits with these resolutions tended to search more than the average visit. The resolution 640×480 has lift less than one. The reason for this is interesting. We found that when the screen resolution was set to 640×480, the search button disappeared past the right edge of the browser screen. In order to access the search button, one would have to scroll to the right, which explains why so few visits with that resolution performed a search.

2. *Weighted averages.* Averaging is a common operation in aggregating data. Although the computation is simple, it is very easy to make mistakes in some situations. In a typical aggregation scenario, *Order Line* (individual line items) data is aggregated to the *Order Header* (a single purchase) level and then to the *Customer* level. Consider, for example, the need to calculate the average amount spent per order line by a customer. The average order line amount per order would be computed in the first aggregation from *Order Line* level to the *Order Header* level by dividing the total order line amount by the number of order lines in the order. However, in the next aggregation from the *Order Header* level to the *Customer* level, it is not correct to simply average the average order line amount per order that was computed previously. Instead, it is necessary to compute a weighted average taking into account the number of order lines in each order placed by the customer. Most transformational tools do not automatically compute averages of averages correctly. We found that by building this knowledge into the tool itself (by having average operations generate weighted numerical attributes rather than simple numerical attributes, and then accordingly taking these weights into account on following average calculations), the chance of user error is reduced.
3. *Visualization.* A picture is worth a thousand words. Visualization tools ranging from elementary *line* and *bar charts* to *scatter plots*, *heatmaps*, and *filter charts* are very useful in identifying interesting trends and patterns in the data.

   Bar charts are used to visualize attribute distributions and to study the degree of correlation between two attributes (see the lift chart in figure 5).

   Scatter plots enable users to visualize the interaction between multiple attributes. Figure 6 shows a scatter plot where recency and frequency are mapped to the X and Y-axes respectively; the size of each square is mapped to the number of customers in that segment; and the color of each square is mapped to the average response spending which ranges from *light gray* (low) to *black* (high). Recency and frequency are ordered from one to five with one being the most recent and most frequent respectively and five being the least recent and least frequent respectively. The more recent and more frequent purchasers have a significantly better response in terms of higher average spending than those who have shopped less recently or infrequently. The scatter plot emphasizes the segment sizes due to attribute interactions. The squares are of different sizes indicating that there are a different number of customers in each segment. For instance, the square corresponding to recency $= 1$ and frequency $= 1$ is larger than its neighbors since it represents a larger segment of the customer base as compared to its neighbors. Similarly, a three-dimensional scatter plot can depict the interactions between five attributes (three assigned to the X, Y, and Z axes respectively, one to color, and one to size).

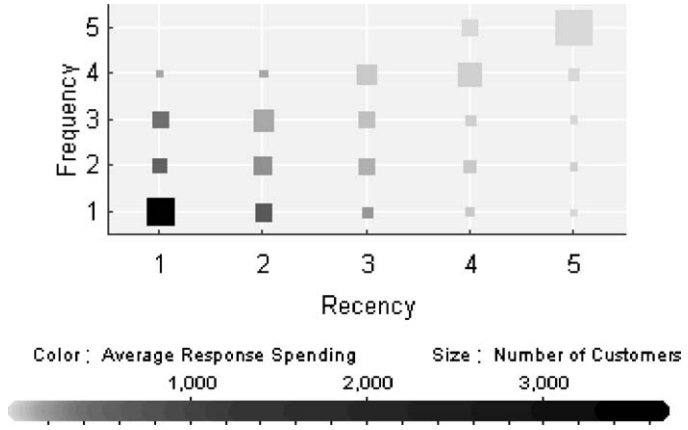


Figure 6. Scatter plot depicting customer segments by recency and frequency.

Heatmaps help to readily discern interesting trends over time. Figure 7 shows the visits to the website over time in the form of a line chart. Without looking at a calendar it is hard to understand that the periodic decrease in traffic is on weekends. Figure 8 shows a heatmap that plots week versus day of week to visualize the traffic patterns for the same time period. Color is mapped to the number of visits that ranges from *light gray* (low) to *black* (high). The heatmap clearly shows that the website attracts fewer visitors during the weekends (Saturday and Sunday). Further, the entire week of December 30, 2002 had low traffic due to the New Year holidays. Monday, February 17, 2003 had lighter traffic compared to other Mondays in the graph owing to the US President's Day holiday. Such patterns are extremely hard to discern from the chart in figure 7.

Filter charts allow users to interactively pick different attributes to filter the data on and immediately see the impact of the selected filter settings. Figure 9 shows a sample filter chart with two attributes: *repeat purchaser* and *tenure* (number of years as a customer). Typical filter charts have between five and ten attributes. Each attribute is depicted as a histogram with the height corresponding to the number of customers. The user can select one or more attribute values by clicking on the corresponding bars and apply these settings to see their effect on other charts. For example, the result of selecting high tenure (*tenure* = 3) and repeat purchaser (*repeat purchaser* = true) from the filter chart on the scatter plot from figure 6 is shown in figure 10. Filter charts can thus be used to effectively identify interesting sub-segments of the customer base. In a practical application of using filter charts to analyse Debenhams' data, we identified a large group of Debenhams' loyalty card members with very high average spending per order but who had not purchased very recently and were not frequent purchasers (Blue Martini Software, 2003b). Note that Debenhams has contact information for these customers since they are loyalty card members. Thus, customers in this group are good candidates for a marketing promotion encouraging them to purchase again.

4. *Enriched customer signatures.* Customers are the center of many data analysis projects in retail e-commerce. To generate good insights and effective models in customer-centric

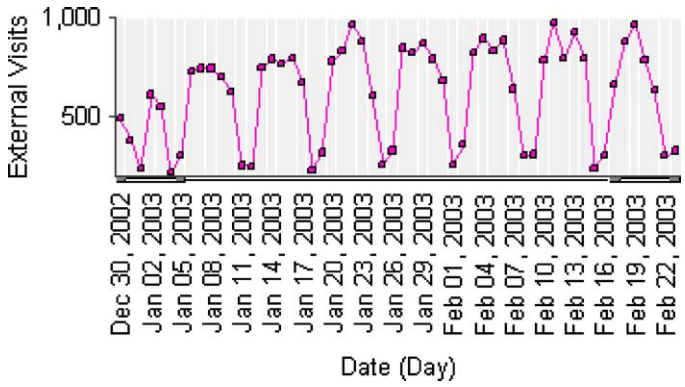


Figure 7. Line chart showing visits to the website over time.

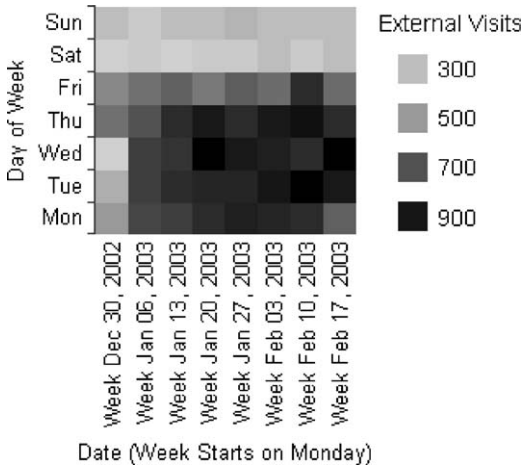


Figure 8. Heatmap showing trends in visits to the website over time.

analyses, it is imperative to generate rich customer signatures covering all aspects of the customers' interactive history with the business. From our experience, the following information should be part of the signature in general: (1) customer registration information, (2) aggregated information from customer web visits including referrers, areas of the site visited, products viewed and purchased, average session length, visit frequency, searches, and abandoned shopping carts and their contents, (3) customer purchase information including Recency, Frequency, and Monetary (RFM) scores (Hughes, 2000), (4) campaign response, (5) performance and error-related logs, (6) bricks-and-mortar store shopping history if available, (7) demographic and socio-economic attributes such as those available from data providers such as Acxiom and Experian. Beyond these standard features, every client data has its own set of custom attributes. These attributes can

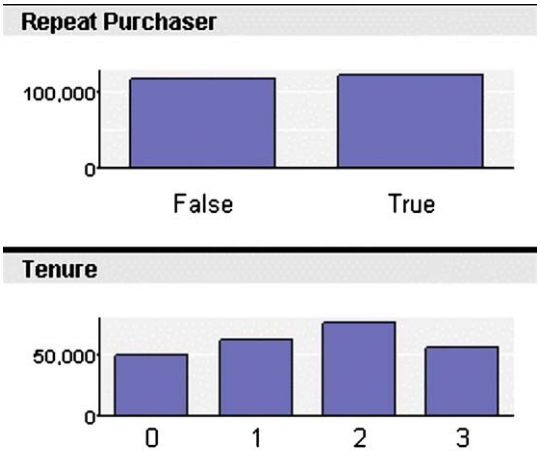


Figure 9. Filter chart depicting two attributes.

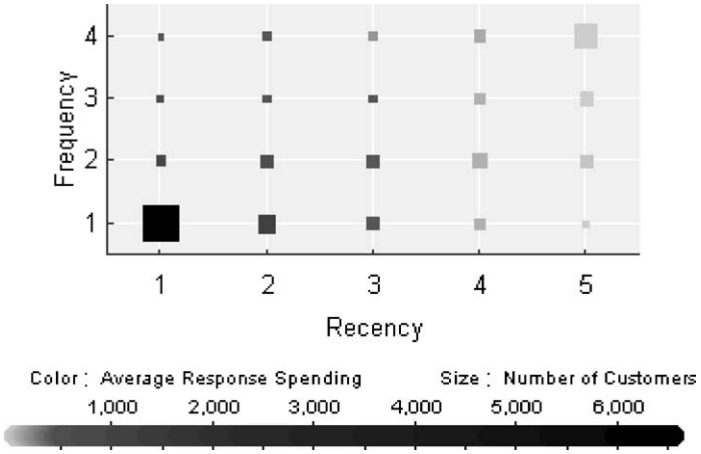


Figure 10. Scatter plot for customer segments filtered to high tenure, repeat purchasers.

be used to further enrich the customer signature. A rich customer signature can serve as the starting point for many interesting analyses such as *migrator* studies and RFM analysis (Blue Martini Software, 2003a).

4.2. Building models and identifying insights

Analyzing data is a complex task (Berry & Linoff, 1997, 2000). It needs experience and expertise. Nevertheless, starting from the questions specified by the business users is a good and practical approach to developing appropriate models. To a data analyst, building

models and identifying novel insights is almost always the more interesting part of the entire analysis project. We share below several lessons we have learned from our experiences in model building.

1. *Mine data at right granularity levels.* Practical data mining scenarios involve data that are collected at different levels of granularity. For example, in retail e-commerce we have *Order Line* level data, *Order Header* level data, and *Customer* level data as explained in Section 4.1. In general, each customer may place one or more orders, and each order may have one or more line items. A common scenario would involve creation of a *star-schema* with the Order Line data as the fact and Order Header and Customer data joined in as dimensions (Kimball, 1996; Kimball et al., 1998; Rosset et al., 1999). The resulting data contains one row per order line with the order header and customer level attributes joined. Computing the number of male and female customers from this data without regard to the granularity levels would give incorrect results because the customer attributes are repeated for each order line that is placed by the customer. The number of males and females must be computed at the correct granularity level (the customer level). Data at different levels of granularity must be suitably aggregated before using it for mining (Pyle, 1999). For instance, to determine the characteristics of heavy spenders the data must first be aggregated to the customer level (Kohavi et al., 2000).
2. *Handle leaks in predictive models.* When building predictive models one has to be careful about potential leaks in the data. Leaks are attributes that are highly correlated with the target but not useful in practice as good predictors. In several cases, these leaks are variables that are derived from the target. For example, in an analysis to characterize heavy spenders, it will be observed that the tax amount is very likely highly correlated with heavy spending in that, the more you spend, the higher is the tax you owe. Other leaks might not be as obvious. For example, the use of free shipping might be correlated to heavy spending. This might be owing to a free shipping promotion when you buy merchandise over $50. Identifying and removing leaks is a tedious process and involves active cooperation between the analyst and the business experts. The problem of leaks is ameliorated to some extent when targets of predictive models are defined based on time-based measures. For example, a *migrator* is defined as a customer who makes small purchases over one time period (say the first year) but migrates to a heavy spender over the next time period. To characterize migrators we identify their characteristics during the earlier year and use these attributes to predict their behavior the following year. A careful selection of attributes used for prediction in this case will reduce the likelihood of leaks. Thus, as opposed to heavy spender analysis that is fraught with potential leaks, it is worthwhile to perform a migrator analysis, which does not suffer as much from the problem of leaks.
3. *Improve scalability.* The need for fast, scalable data mining algorithms has been recognized since the early days of data mining research (Chan & Stolfo, 1997; Freitas, 1998; Freitas & Lavington, 1998; Provost & Kolluri, 1999). The tremendous advances in computing power and systems resources have been matched by the increase in the quantity of data being analyzed. Sites that have 30 million page views per day will need to house about 10 billion records each year. Pfahringer (2002) wrote "in [KDD Cup] 1999 for

instance, I was never able to run learning algorithms on the full dataset. Only sub-samples were practical in terms of processing time or main memory limits." Sampling has been used quite effectively as a way to address the data size problem (Domingos, 2002). It is worth mentioning that sampling should be performed at the correct granularity level. For example, in the retail e-commerce arena, sampling should be performed at the customer (or visitor) level. A certain portion of all customers must be selected at random along with all order data and all clickstream data corresponding to these customers. If sampling is performed at the clickstream level instead, then the resulting sample will include a random subset of all clicks (page requests) from which a clear picture of the customers purchasing behavior and navigational preferences cannot be constructed. Parallel and distributed processing is another effective way of reducing the computation time. While parallel versions of the popular data mining algorithms have received a lot of attention (Agrawal & Shafer, 1996; Maniatty & Zaki, 2000), several practical challenges still remain unsolved. One example is the inability of most data mining algorithms to scale to thousands of attributes. Pfahringer (2002) wrote "In [KDD Cup] 2001, none of the standard tools, not even the commercial ones, could directly deal with the training set of about 2000 examples comprised of approximately 140,000 attributes each."

4. *Build simple models first.* A common mistake often made by analysts involves beginning the analysis without establishing a baseline to compare the results of analysis. The temptation to build powerful models using the most sophisticated tools should be resisted at least until a clear need is established for these models. Business users tend to appreciate and accept models that are more understandable. Further, the business might have some sort of model already in place. In order to convince the business user that the current model must be replaced, the analyst must start from the existing model as the baseline and work on approaches to improve upon it. Projects where the business users are not comfortable with the models that have been built stand the risk of not being implemented at all. It is therefore a good practice to start simple, earn the confidence of the business user, and then gradually develop more sophisticated models as necessary. In a recent analysis of a client's data, we were required to identify interesting segments within their customer base. The approach we took was to build a data cube based on recency, frequency, and monetary scores that could be easily visualized using simple visualizations. RFM analysis has been the workhorse of marketers for several decades and has been used extensively for customer segmentation (David Shepard Associates, 1998; Hughes, 2000). Additional customer attributes such as demographics, browsing behavior, and purchasing history were used to augment the cube. The business users at the client side who happened to be marketing experts were extremely pleased to see a representation of the data that they understood and could interactively explore using visualization tools such as filter charts described in Section 4.1. We found that the client was much more receptive to our recommendations and appreciative of the efforts we had put in.
5. *Use data mining suites.* There are several advantages of using data mining suites where comprehensive tools for data processing, transformation, cleansing, analysis, and reporting are available in a single package. Such suites simplify data processing, analysis, and closing the loop, as there is no need to transfer data between disparate systems. Further, the availability of different types of tools enables the analysts and business users to pick

the tools that are most appropriate for the type of analysis they are performing. Designers of commercial data mining software claimed to have taken this approach from the very beginning. Often however, these suites have several data mining algorithms (Elder & Abbot, 1998) but lack capabilities such as data transformations, or reporting, or visualization.

6. *Peel the onion and validate results.* Langley addressed the need to move beyond describing the data to providing explanations of the data (Langley, 2002). However, extreme care must be taken while providing explanations of the data. Often during analysis we are likely to come up with superficial correlations that seem interesting and also make perfect business sense. Careful investigation may reveal very different relationships hidden beneath. It is therefore imperative to peel the onion and ascertain the true merit of the proposed explanation of the data. A good example is from the KDD Cup 2000 competition question 3: "*Characterize heavy spenders at Gazalle.com web store*" (Kohavi et al., 2000). Several submissions contained the rule "*Customers who expressed willingness to receive emails from Gazalle.com are heavy spenders*." At first sight, we found this makes sense as it reflects the customers' loyalty. However, this was primarily due to the correlation between time and the willingness to receive emails. Gazelle.com changed the default for the registration question about receiving emails from *yes* to *no* on 2/28 and back to *yes* on 3/16. The resulting changes in the percentage of customers who are willing to receive emails are evident in figure 11. Independent of this change, a huge promotion was announced on 2/28 that offered a $10 discount off every purchase. This drove down the percentage of heavy spenders during the promotion period which happened by chance to coincide with the duration for which the default value for the registration question regarding receiving emails was *no*.

4.3. *Sharing insights, deploying models, and closing the loop*

The ultimate objective of most data analysis projects is to use the insights and the models to improve business. The analysis is incomplete until the results of the analysis are shared

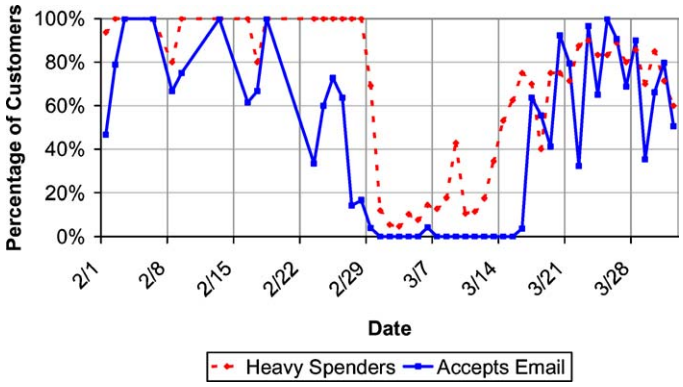


Figure 11. Correlation between heavy spending and willingness to receive emails.

Figure 12. A visualization of Naïve-Bayes.

across the relevant departments in the organization and concrete actions have been taken based on the findings.

1. *Represent models visually for better insights.* Generated models and insights are much better understood when presented in visual format. Simple visual representations such as bar charts, line charts, and heatmaps can convey a lot of information in a concise yet effective manner. Business users do not want to deal with advanced statistical concepts. They want straightforward visualizations and task-relevant outputs. Consider figure 12, which summarizes a Naïve-Bayes model for predicting which people earn more than $50,000 in yearly salary. Instead of the underlying log conditional probabilities that the model actually manipulates, the visualization uses bar height to represent evidence for each value of a contributing factor listed on the left of the figure and color saturation to signify confidence of that evidence (Becker, Kohavi, & Sommerfield, 2001). For example, evidence for higher salaries increases with age, until the last age bracket, where it drops off; evidence for higher salaries increases with years of education, with the number of hours worked, and with certain marital statuses (e.g., married to civilian spouse, but not married to a spouse in the armed forces) and occupations (e.g., professional specialty and executive managerial). Note also that the visualization shows only a few attributes that were determined by the mining algorithm to be the most important ones, highlighting to the business users the most critical attributes from a larger set.
2. *Understand the importance of the deployment context.* It is common in practice to have the analyst or data mining team develop the models and have the marketing or the IT team deploy these models (this often happens in an interactive and iterative fashion). In order to ensure a successful deployment of the designed models it is very important to understand the deployment context. For example, a product recommender model based on association rules might not be deployed in its entirety. The marketers responsible for manually picking up cross-sells and up-sells might only want to deploy those rules that make sense to them or might want to merge their hand crafted rules with the generated model. In this case, it is important that the generated model be editable. In another project, we came upon a physical limitation imposed by the mail house that is responsible for printing and mailing physical letters. The client performs segmentation of their

customers based on purchase history and propensity to purchase. Multiple monthly campaigns are run based on this segmentation. Each campaign results in an insert for the monthly mailing sent to the customer. The mail house they use has a physical limitation that the different letters belonging to a segment could only be coded by a two-digit code. As a result, the above segmentation process was limited to finding at most 100 distinct segments. This poses a problem especially when the client wants to run up to 30 campaigns each month, which can potentially create 2^{30} segments.

3. *Creating actionable models and closing the loop.* Insights and models that are directly actionable are usually more interesting and can directly impact the business. Consider the insight "*Heavy spenders tend to purchase blue shirts*." This might be a good insight but is not readily actionable because it is not clear whether a visitor to the website is a heavy spender or not. On the other hand, an insight like "*Visitors referred by Google tend to purchase blue shirts*." It is easy to determine in real-time whether the visitor to the site is referred from Google and take appropriate action such as promote the latest blue shirts to the visitor. Some models might be actionable but might require complex processing. As a result these models are not readily deployable for use in real-time say at the live website. Finally, it is preferable to develop systems whereby models can be automatically updated with little or no manual intervention. In our system, a product recommender or scoring model can be updated nightly or weekly based on the new data and deployed automatically to the website to help in targeting new visitors.

4.4. Empowering business users to conduct their own analyses

A big challenge for data mining is the need to reduce the time and analytical expertise required to analyze the data. Empowering business users to perform their own analysis would partly address this issue. At the same time, we must mention that expert analysts must perform the more involved analysis. This is to prevent misinterpretation of the results by a non-expert, which can prove to be more costly in the long run.

1. Share the results among business users via simple, easy to understand reports. A web browser accessible business intelligence portal is quickly becoming the method of choice for sharing the insights.
2. Provide canned reports that can be run by business users by simply specifying values for a few parameters such as date ranges. This provides a method to allow business users to change the behaviors of data mining analysis in controlled ways.
3. Technically savvy business users might be comfortable designing their own investigations if a simple graphical user interface is provided as part of the data mining tool.

A variety of challenges related to performing the analysis remain to be addressed:

1. *Visualize models.* More sophisticated models such as rules and associations are not easy to visualize. Despite some recent advances in this area (Zhang, 2000) a suitable visualization approach that will help business users clearly understand the model is yet to be designed.

2. *Prune rules and associations.* Often a product recommender model comes up with thousands or even millions of association rules (Zheng, Kohavi, & Mason, 2001). It is impossible for a human to manually go through even a small number of these rules. Further, scoring based on such complex models is an expensive operation and might slow down the operations at the website where the model is deployed. Some work on generating only top-N rules with respect to certain criterion (Webb, 2000) might help to some extent.
3. *Analyze and measure long-term impact of changes.* Business actions such as promotions, altered procedures, changes to the user experience, etc. have short and long term impact. Often, the short-term impact of an action is easy to measure. For example, test and control groups can be used to identify the impact of a business action in the short term. Long-term impact is much more difficult to analyze and measure. For instance, an email promotion might boost sales in the short term but in the long run frequent email blasts might encourage customers to opt-out from the email list. Frequent promotional campaigns might also result in an undesired effect. Customers who get used to receiving these frequent promotions hold off purchasing products until the next promotion is announced. Retailers have overcome this problem to some extent by not pre-announcing promotions and designing promotions such that they have very short expiry durations.

5. Summary

We reviewed the Blue Martini Software architecture, which provided us with powerful capabilities to collect additional clickstream data not usually available in web logs, while also obviating the need to solve problems usually bottlenecking analysis (and which are much less accurate when done as an afterthought), such as sessionization and conflating data from multiple sources. We believe that such architectures where clickstreams are logged by the application server layer are significantly superior and have proven themselves with our clients and at other sites like Amazon.com, which uses a proprietary application server.

Our focus on Business to Consumer (B2C) e-commerce for retailers allowed us to drill deeper into business needs to develop the required expertise and design out-of-the-box reports and analyses in this domain. Further, we believe that most lessons and challenges will generalize to other domains outside of retail e-commerce.

We reviewed many lessons at differing levels of granularity. If we were to choose the top three, they would be:

1. Integrate data collection into operations to support analytics and experimentation, and make it easy to transfer the collected information to a data warehouse where it can be combined and conflated for a 360-degree view.
2. Do not confuse yourself with the target user. Provide as much insight out-of-the-box and make it easy to derive insight and take action. While this is certainly easier in a specific domain, we believe it is the only way to succeed in businesses that have little analytical expertise in-house. Business users have a daily job to perform and learning about data mining is not on the top of their agenda. However, insight that can impact their decisions and help them optimize the business is certainly ranked high.

3. Provide simple reports and visualizations before building more complex models. Many of the strongest insights are usually a side effect of a business process. If something looks too good to be true, or has too high a confidence, you must peel the onion and drill deeper keeping in mind that correlation does not imply causality.

For challenges, the top three would be:

1. The ability to translate business questions to the desired data transformations is especially hard.
2. Efficient algorithms whose output is comprehensible for business insight, and which can handle multiple data types (dates, hierarchical attributes, different granularity data) need to be designed.
3. Integrated workflow. Many business tasks require multiple people and processes (Chapman et al., 2000). More guidance and tracking of progress is necessary.

The web is an experimental laboratory (Kohavi, 2001) where hundreds of experiments can be performed easily and quickly, but data must be collected for reasons other than operational performance and debugging (the main reasons for standard web logs).

E-commerce is still in its infancy, with less than a decade of experience. Best practices and important lessons are being learned every day. The Science of Shopping (Underhill, 2000) is well developed for bricks and mortar stores. Despite our success with the Blue Martini architecture, there is significant work remaining in understanding data mining in the context of retail e-commerce.

Acknowledgments

We thank the members of the data mining team at Blue Martini Software for the numerous discussions and debates that have helped to shape the ideas in this paper. We are grateful to our clients for sharing their data with us. We thank the editors and the anonymous reviewers for their insightful comments and suggestions on improving the paper. We are also grateful to numerous people who helped us with feedback, including Jon Becher, Tom Breur, Rob Cooley, Rob Gerritsen, David Liu, Brij Masand, Foster Provost, Ross Quinlan, Paat Rusmevichientong, David Selinger, Evangelos Simoudis, Jim Sterne, Kai Ming Ting, Noe Tuason, Alex Tuzhilin, Geoff Webb, Andreas Weigend, and Shenghuo Zhu.

References

ANSI/X3/SPARC. (1975). Study group on data base management systems. Interim Report, ANSI.

Almuallim, H., Akiba, Y., & Kaneda, S. (1995). On handling tree-structured attributes. In *Proceedings of the Twelfth International Conference on Machine Learning (ICML'95)* (pp. 12–20). Morgan Kauffmann.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94)* (pp. 487–499). Morgan Kauffmann.

Agrawal, R., & Shafer, J. (1996). Parallel mining of association rules. *IEEE Transactions of Knowledge and Data Engineering, 8*, 962–969. IEEE. http://www.almaden.ibm.com/cs/people/ragrawal/papers/parassoc96.ps.

Ansari, S., Kohavi, R., Mason, L., & Zheng, Z. (2001). Integrating E-commerce and data mining: Architecture and challenges. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'2001)*. IEEE. http://www.lsmason.com/papers/ICDM01-eCommerceMining.pdf.

Aronis, J., & Provost, F. (1997). Increasing the efficiency of data mining algorithms with breadth-first marker propagation. In *Proceedings of Knowledge Discovery and Data Mining (KDD'97)* (pp. 119–122). AAAI Press.

Becker, B., Kohavi, R., & Sommerfield, D. (2001). Visualizing the simple Bayesian classifier. *Information Visualization in Data Mining and Knowledge Discovery, 18*, 237–249. Morgan Kaufmann. http://robotics.stanford.edu/users/ronnyk/ronnyk-bib.html.

Berry, M., & Linoff, G. (1997). *Data mining techniques: For marketing, sales, and customer support*. John Wiley and Sons.

Berry, M., & Linoff, G. (2000). *Mastering data mining: The art and science of customer relationship management*. John Wiley and Sons.

Blue Martini Software. (2003a). Blue Martini business intelligence at work: Charting the terrains of MEC Website data. http://robotics.stanford.edu/users/ronnyk/ronnyk-bib.html.

Blue Martini Software. (2003b). Blue Martini business intelligence delivers unparalleled insight into user behavior at the Debenhams Web site. http://robotics.stanford.edu/users/ronnyk/ronnyk-bib.html.

Catledge, L., & Pitkow, J. (1995). Characterizing browsing strategies in the World-Wide Web. *Computer Networks and ISDN Systems, 27:6*, 1065–1073. Elsevier Science. http://citeseer.ist.psu.edu/catledge95characterizing.html.

Chan, P., & Stolfo, S. (1997). On the accuracy of meta-learning for scalable data mining. *Journal of Intelligent Information Systems, 8:1*, 5–28. Kluwer Academic Publishers. http://www1.cs.columbia.edu/∼pkc/papers/jiis97.ps.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Sherer, C., & Wirth, R. (2000). Cross industry standard process for data mining (CRISP-DM) 1.0. http://www.crisp-dm.org/.

Cheswick, W., & Bellovin, S. (1994). *Firewalls and internet security: Repelling the wily hacker*. Addison-Wesley Publishing Company.

Cohen, W. (1996). Learning trees and rules with set-valued features. In *Proceedings of the AAAI/IAAI Conference, 1*, 709–716. AAAI Press.

Collins, J., & Porras, J. (1994). *Built to last, successful habits of visionary companies*. Harper Collins Publishers.

Cooley, R., Mobasher, B., & Srivastava, J. (1999). Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems, 1:1*. Springer-Verlag. http://maya.cs.depaul.edu/∼mobasher/papers/webminer-kais.ps.

David Shepard Associates. (1998). *The new direct marketing: How to implement a profit-driven database marketing strategy*, 3rd edition. McGraw-Hill.

Domingos, P. (2002). When and how to subsample: Report on the KDD-2001 panel. *SIGKDD Explorations, 3:2*, 74–76. ACM. http://www.acm.org/sigs/sigkdd/explorations/issue3-2/contents.htm#Domingos.

Elder, J., & Abbott, D. (1998). A comparison of leading data mining tools. *Tutorial at the Knowledge Discovery and Data Mining Conference (KDD'98).* ACM. http://www.datamininglab.com/pubs/kdd98_elder_abbott_nopics_bw.pdf.

English, L. (1999). *Improving data warehouse and business information quality: Methods for reducing costs and increasing profits*. John Wiley & Sons.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (Eds.). (1996). *Advances in knowledge discovery and data mining*. MIT Press.

Freitas, A. (1998). Tutorial on scalable, high-performance data mining with parallel processing. In *Proceedings of the Principles and Practice of Knowledge Discovery in Databases (PKDD'98).* Springer.

Freitas, A., & Lavington, S. (1998). *Mining very large databases with parallel processing*. Kluwer Academic Publishers.

Heaton, J. (2002). *Programming spiders, bots, and aggregators in Java*. Sybex Book.

Hughes, A. (2000). *Strategic database marketing*, 2nd edition. McGraw-Hill.

Kimball, R. (1996). *The data warehouse toolkit: Practical techniques for building dimensional data warehouses*. John Wiley & Sons.

Kimball, R., & Merz, R. (2000). *The data webhouse toolkit: Building the Web-enabled data warehouse*. John Wiley & Sons.

Kimball, R., Reeves, L., Ross, M., & Thornthwaite, W. (1998). *The data warehouse lifecycle toolkit : Expert methods for designing, developing, and deploying data warehouses*. John Wiley & Sons.

Kohavi, R. (1998). Crossing the Chasm: From academic machine learning to commercial data mining. *Invited talk at the Fifteenth International Conference on Machine Learning (ICML'98)*, Madison, WA. Morgan Kauffmann. http://robotics.stanford.edu/users/ronnyk/ronnyk-bib.html.

Kohavi, R. (2001). Mining e-commerce data: The good, the bad, and the ugly. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2001)* (pp. 8–13). ACM Press. http://robotics.stanford.edu/users/ronnyk/ronnyk-bib.html.

Kohavi, R., Brodley, C., Frasca, B., Mason, L., & Zheng, Z. (2000). KDD-Cup 2000 organizers' report: Peeling the onion. *SIGKDD Explorations, 2:2*, 86–98. ACM Press. http://robotics.stanford.edu/users/ronnyk/ronnyk-bib.html.

Kohavi, R., & Provost, F. (2001). Applications of data mining to electronic commerce. *Data Mining and Knowledge Discovery, 5:1/2*. Kluwer Academic. http://robotics.Stanford.EDU/users/ronnyk/ecommerce-dm.

Kohavi, R., Rothleder, N., & Simoudis, E. (2002). Emerging trends in business analytics. *Communications of the ACM, 45:8*, 45–48. ACM Press. http://robotics.stanford.edu/users/ronnyk/ronnyk-bib.html.

Langley, P. (2002). Lessons for the computational discovery of scientific knowledge. *Proceedings of the First International Workshop on Data Mining Lessons Learned (DMLL'2002).* http://www.hpl.hp.com/personal/Tom_Fawcett/DMLL-2002/Langley.pdf.

Lee, J., Podlaseck, M., Schonberg, E., & Hoch, R. (2001). Visualization and analysis of clickstream data of online stores for understanding Web merchandising. *Data Mining and Knowledge Discovery, 5:1/2*. Kluwer Academic.

Linoff, G., & Berry, M. (2002). *Mining the Web: Transforming customer data*. John Wiley and Sons.

Madsen, M. R. (2002). Integrating Web-based clickstream data into the data warehouse. DM Review, August, 2002. http://www.dmreview.com/editorial/dmreview/print_action.cfm?EdID=5565.

Maniatty, W., & Zaki, M. (2000). A requirements analysis for parallel (KDD) systems. In *Proceedings of the Data Mining Workshop at the International Parallel and Distributed Processing Symposium (IPDPS'2000)*. IEEE Computer Society.

Mason, L., Zheng, Z., Kohavi, R., & Frasca, B. (2001). Blue Martini eMetrics study. http://developer.bluemartini.com.

McJones, P. (1995). The 1995 SQL reunion: People, projects, and politics an informal but first-hand account of the birth of SQL, the history of System R, and the origins of a number of other relational systems inside and outside IBM. http://www.mcjones.org/System_R/SQL_Reunion_95/sqlr95-System.html.

Pfahringer, B. (2002). Data mining challenge problems: Any lessons learned? In *Proceedings of the First International Workshop on Data Mining Lessons Learned (DMLL'2002).* http://www.hpl.hp.com/personal/Tom_Fawcett/DMLL-2002/Proceedings.html.

Piatetsky-Shapiro, G., Brachman, R., Khabaza, T., Kloesgen, W., & Simoudis, E. (1996). An overview of issues in developing industrial data mining and knowledge discovery applications. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)* (pp. 89–95). AAAI Press.

Provost, F., & Kolluri, V. (1999). A survey of methods for scaling up inductive algorithms. *Data Mining and Knowledge Discovery, 3:2*, 131–169. Kluwer Academic.

Pyle, D. (1999). *Data preparation for data mining*. Morgan Kauffmann.

Quinlan, R. (1986). Induction of decision trees. *Machine Learning, 1*, 81–106. Kluwer Academic.

Quinlan, R. (1989). Unknown attribute values in induction. In *Proceedings of the Sixth International Machine Learning Workshop (ICML'89)* (pp. 164–168). Morgan Kauffmann.

Rosset, S., Murad, U., Neumann, E., Idan, Y., & Pinkas, G. (1999). Discovery of fraud rules for telecommunications: Challenges and solutions. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99)* (pp. 409–413). ACM Press. http://www-stat.stanford.edu/%7Esaharon/papers/fraud.pdf.

RuleQuest Research. (2003). C5.0: An informal tutorial. http://www.rulequest.com/see5-unix.html.

Simpson, E. (1951). The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society, Ser. B, 13*, 238–241.

Spiliopoulou, M., Mobasher, B., Berendt, B., & Nakagawa, M. (2003). A framework for the evaluation of session reconstruction heuristics in Web usage. *INFORMS Journal of Computing, Special Issue on Mining Web-Based Data for E-Business Applications, 15:2*. http://maya.cs.depaul.edu/~mobasher/papers/SMBN03.pdf.

Tan, P., & Kumar, V. (2002). Discovery of Web Robot sessions based on their Navigational patterns. *Data Mining and Knowledge Discovery, 6:1*, 9–35. Kluwer Academic. http://www-users.cs.umn.edu/~ptan/Papers/DMKD.ps.gz.

Underhill, P. (2000). *Why we buy: The science of shopping*. Touchstone Books.

Webb, G. I. (2000). Efficient search for association rules. In *Proceedings of the Discovery and Data Mining Conference (KDD 2000)* (pp. 99–107). ACM Press. http://portal.acm.org/citation.cfm?id=347112&coll=portal&dl=portal&CFID=8086514&CFTOKEN=81282849.

Zhang, H. (2000). Mining and visualization of association rules over relational DBMSs. PhD thesis, Department of Computer and Information Science and Engineering, The University of Florida. http://citeseer.ist.psu.edu/cache/papers/cs/20450/http:zSzzSzetd.fcla.eduzSzetdzSzufzSz2000zSzana 7033zSzEtd.pdf/zhang00mining.pdf.

Zhang, J., Silvescu, A., & Honavar, V. (2002). Ontology-driven induction of decision trees at multiple levels of abstraction. In *Proceedings of Symposium on Abstraction, Reformulation, and Approximation. Lecture Notes in Artificial Intelligence* (Vol. 2371), Springer-Verlag.

Zheng, Z., Kohavi, R., & Mason, L. (2001). Real world performance of association rule algorithms. In *Proceedings of the Knowledge Discovery and Data Mining Conference (KDD 2001)* (pp. 401–406). ACM Press. http://www.lsmason.com/papers/KDD01-RealAssocPerformance.pdf.

Received April 7, 2003
Accepted April 8, 2004
Final manuscript April 13, 2004