Communications of the Association for Information Systems

Volume 22 Article 33

4-2008

You've Data Mined. Now What?

Michael Brydon
Simon Fraser University

Andrew Gemino
Simon Fraser University, gemino@sfu.ca

Recommended Citation

Brydon, Michael and Gemino, Andrew (2008) "You've Data Mined. Now What?," *Communications of the Association for Information Systems*: Vol. 22 , Article 33.
DOI: 10.17705/1CAIS.02233
Available at: https://aisel.aisnet.org/cais/vol22/iss1/33

Communications of the Association for Information Systems

CAIS

You’ve Data Mined. Now What?

Michael Brydon,

Andrew Gemino

Faculty of Business Administration

Simon Fraser University

gemino@sfu.ca

Abstract:

Data-mining technologies are within the grasp of many organizations. Commercially available data-mining packages make it relatively easy for firms to transform their data resources into predictive models. Yet, despite technological advances, the precise manner in which data-mining output should be incorporated into an organization’s decision-making processes remains unclear. This paper attempts to clarify the role of data mining by situating it within the context of Simon’s model of decision making. We use a complex decision problem from the video game development industry to illustrate several practical challenges managers face when using data-mining output as a decision making input. We then show how some of these challenges can be overcome by incorporating data-mined predictive models into a conventional decision-analytic formulation of the problem.

KEYWORDS: Data mining, decision support, video game development, real options, decision-theoretic planning

Volume 22, Article 33, pp.603- 616, June 2008

I. INTRODUCTION

A byproduct of every business process is information about how the process can be improved [Box 1957]. Some firms, such as Amazon and Harrah’s, collect large amounts of data from their routine operations and have demonstrated the ability to make decisions based on insights extracted from this data [Davenport 2006; Loveman 2003]. Many firms, however, have been less successful transforming their data resources into better decisions. A major grocery chain, for example, reported using less than two percent of the scanner data that it had amassed over the years [Davenport 2001]. The failure of some firms to exploit data for decision making is surprising given that the prerequisite technologies for sophisticated data analysis are often either already in place or are readily available [Hannula and Pirttimaki 2003; Negash 2004].

One data-analysis technology that has attained the maturity required for widespread business use is data mining [Jackson 2002]. Commercial data-mining packages from SAS, SPSS, IBM, and others implement sophisticated algorithms developed over several decades in the data-mining and machine-learning research communities [Haughton et al. 2003]. These algorithms rely on statistical or information-theoretic heuristics to identify possible relationships between variables in large data sets. In many of the applications reported in the literature, data-mining software is used for clustering tasks, such as customer segmentation and basket analysis [Apte et al. 2002]. However, data mining can also be used for predictive modeling, in which stronger assumptions are made about causality in the mined relationships. For example, a model that predicts sales based on variations in product attributes implicitly assumes that certain product attributes *cause* sales.

The effectiveness of data-mining algorithms for identifying possible causal relationships has been well established in the data-mining literature [Jackson 2002]. However, as Alter [2004] points out:

> *Decision support is not about tools per se, but rather, about making better decisions within work systems in organizations. The common emphasis on features and benefits of DSS as artifacts rather than on how to improve decisional aspects of work systems in organizations may contribute to the frequently cited and occasionally questioned failure rates of […] technology-based innovations.*

The objective of this paper is to examine the role of data mining within the larger context of decision making and to identify some of the challenges that firms face as they attempt to transform their data resources into better decisions. We use an example problem from the video game development industry to demonstrate the relative ease with which a predictive model can be created using commercial data-mining software and third-party data. However, we also use the video game example to illustrate the mismatch that arises in many organizations between the techniques used to make predictions and those used to make decisions. We conclude that firms must be willing to reconsider their overall decision-making processes if they hope to reap benefits from their investments in data mining.

II. THE DECISION MAKING PROCESSES

In the summer of 2005, decision makers at GameCorp, a major video game development company, faced a troubling statistic. The firm could spend upwards of $17M to develop a new blockbuster game; however, the median revenue generated by a console-based video game was only $3M. Of the thousands of new games released each year, only about 200 manage to recoup their development costs [Gaume 2006]. The economic risk is amplified in the video game industry by the long lag between development decisions and the market’s response to those decisions. Critical elements of a game—such as its genre, target audience, complexity, brand, and licensing—must be decided on early in the game’s development lifecycle even though the implications of these decisions are not observed until the game is on the shelves of retailers. As a consequence, GameCorp and its competitors must make important development decisions based on predictive models—typically implicit—of the market’s response.

Several models of how firms should make decisions in the face of uncertainty have been proposed. One of the best known is Simon’s [1977] intelligence, design, choice model, which decomposes the decision process into three distinct phases. In the intelligence phase, the decision maker searches the environment for conditions calling for a decision. In the design phase, the decision maker invents and develops possible courses of action that might provide an appropriate response to environmental conditions. In the choice phase, the alternatives are evaluated with respect to the decision maker’s objectives and the best course of action is selected. Other models provide a similar decomposition. For example, the sense, model, plan, act (SMPA) model has been widely used within artificial intelligence (AI) to guide the design of decision-making robots [Brooks 1991]. Similarly, the acquisition, evaluation, and judgment phases identified by Einhorn and Hogarth [1981] correspond closely with Simon's model; however, their model emphasizes a fourth phase, learning, which occurs over multiple iterations of the decision process.

As Simon pointed out, the central difficulty with any normative decision model is that human decision makers are only boundedly rational [Simon 1955]. There are clear limits to the ability of humans to identify and evaluate all possible alternatives and thus decision makers must "satisfice" rather than compute an optimal solution. The best that a human decision maker can do is choose an adequate solution from an incomplete set of approximately evaluated alternatives. The problem with satisficing is that our ability to approximate is characterized by several well-known biases [see Arnott 2006]. A central motivating assumption of data mining is that predictive models induced from large amounts of historical data can help push back the bounds of bounded rationality and help decisions makers make better decisions in complex, uncertain environments.

Before taking a closer look at this assumption, we must first define the scope of a typical data-mining analysis and identify the key functionality provided by data-mining tools. Several groups within the data-mining community have proposed process models or methodologies that summarize both scope and functionality [Jackson 2002]. The SEMMA methodology, which is promoted and supported by SAS, a software vendor, identifies five tool-supported tasks (sample, explore, modify, model, and assess) required to transform raw data into a validated predictive model. SPSS, also a vendor, uses the Five As methodology (ask, advise, assess, assist, and arrange) which is similar to SEMMA in scope and purpose. The CRISP-DM process model for data mining, which was developed by a consortium of data-mining vendors and users under the auspices of the European Union's ESPRIT program, is more comprehensive than either the SEMMA or Five As model [Chapman et al. 2000]. CRISP-DM consists of six phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment. We illustrate the six phases of CRISP-DM using our data-mining analysis of video game sales data in Section III. Then, in Section IV, we map the phases of the data-mining analysis to the phases of Simon's model in order to identify tasks that can and cannot be automated, supported, or otherwise improved using data mining tools.

III. PREDICTING BLOCKBUSTERS

The purpose of this section is to provide a high-level summary of the phases of the CRISP-DM process model while simultaneously illustrating the construction of a predictive model for a complex decision problem using off-the-shelf data and data-mining tools. Each subsection following corresponds to a phase of CRISP-DM.

Business Understanding

The first phase of CRISP-DM, business understanding, forces the data-mining analyst to think about the overall objectives of the analysis. In this case, the overall decision problem was to select a set of video game attributes that would maximize the game's expected profitability. Given the long lag between development decisions and the market's response to the game, the specific objective of the data mining analysis was to construct a predictive model of video game sales based on historical data.

Data Understanding

The data used to construct the predictive model was purchased by GameCorp from a well-known market research firm that monitors sales across several industries. The data set contained information about video games released in North America between 2000 and 2004 for the three dominant game platforms during that period: Sony PlayStation 2, Microsoft XBox, and Nintendo GameCube. Although the data set contained only 1,317 unique records (which is small by data-mining standards), it was regarded within the industry as the most comprehensive source of information about the entire population of games released during the period.

The candidate explanatory variables in the data included game genre, brand, target platforms, ESRB rating,[1] license, and release date as well as early indicators of game quality, such as average review score (from a panel of video game critics).

Data Preparation

As in most real-world data sets, the video game data had numerous data quality problems. In addition, it lacked information about each game's competitive environment and its relationship to previously released games of the

[1] The Entertainment Software Rating Board (ESRB) is an industry group that assigns ratings (e.g., "early childhood," "mature," "adults only") to video and computer games.

same brand. We derived additional attributes for each game such as a competition index (the number of games of the same genre that were released in the same year) and the sales performance of the game's prequels (if any). The expanded data set contained 24 candidate explanatory variables.

Life-to-date (LTD) sales of a game across all platforms was chosen as the response variable for the predictive model. Since classification trees require a categorical response variable, each game was assigned to one of three sales categories. The thresholds for the three categories were selected so that the training set was balanced (roughly the same number of games in each category). The average revenues for games in the high, medium, and low categories were $27.8M, $8.7M, and $3.5M respectively. These values reflect the skewness of the underlying sales data.

Modeling

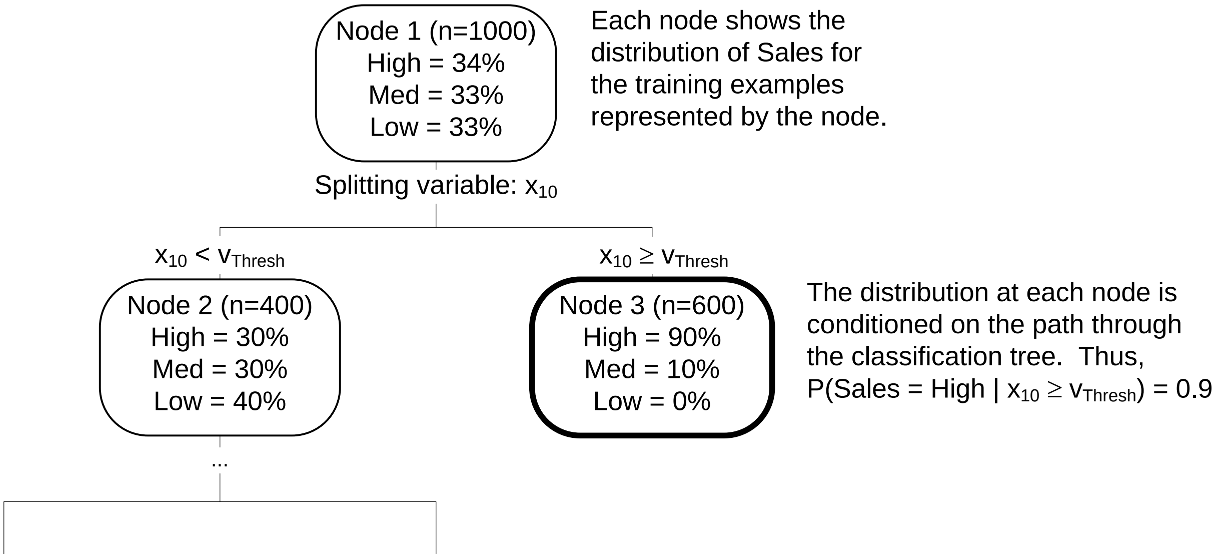
There are a number of techniques GameCorp could have used to develop a predictive model of sales, including linear regression and classification trees. In either case, the algorithm starts with a set of training examples of the form $\langle y, \mathbf{x} \rangle$, where y is the response variable and $\mathbf{x}$ is a vector of game attributes. In linear regression, the predictive model takes the form of an equation in which the unobserved response variable $\hat{y}$ is estimated as a function of the observable variables, $\mathbf{x}$, and the vector of learned coefficients B: $\hat{y} = \mathrm{B}\mathbf{x}$. The uncertainty in the model is expressed as residual variance—the proportion of variation in the response variable that is not explained by the regression equation.

Classifications trees, in contrast, use a graphical representation of the relationships between variables. To create a classification tree, the data-mining algorithm recursively partitions the training data in an effort to create subsets that are as homogeneous as possible with respect to the categorical response variable. To illustrate, consider the simple classification tree in Figure 1. The root node of the tree shows the frequency distribution of sales for the 1,000 video games in the training subset. Commercial data mining packages use a combination of well-established splitting heuristics and brute-force computation to determine which variables in $\mathbf{x}$ best explain variation in the response variable [Kohavi et al., 2002]. Assume in this example that x_{10} was chosen as the first splitting variable. The data in the root node has been partitioned into two subsets according to value of x_{10} relative to a threshold value, v_{Thresh}, which can be set by the data mining analyst or selected by the algorithm. The leaf node on the left branch corresponds to the 400 games for which $x_{10} < v_{Thresh}$ whereas the leaf node on the right branch corresponds to the 600 games for which $x_{10} \geq v_{Thresh}$. As the tree shows, games in the training data with a higher value of x_{10} did better than those with lower values. That is, $P(y = \text{high} \mid x_{10} \geq v_{Thresh}) > P(y = \text{high} \mid x_{10} < v_{Thresh})$. These conditional probabilities suggest that variable x_{10} has been an important predictor of successful games. However, the distribution of sales outcomes when $x_{10} < v_{Thresh}$ provides roughly the same information as the root node. The algorithm would therefore attempt to further partitioning this node until either all examples belonged to the same class or a stopping criterion (such as a minimum number of examples at a node) was encountered.

The result of recursive partitioning is a relatively coarse-grained but concise and easily understood graphical predictive model. The probability of a new game achieving low, medium, or high sales can be estimated by traversing the tree based on the known values of the game's explanatory variables. In addition, the uncertainty associated with each branch of the classification tree is explicit in the leaf node's conditional probability distribution.

SAS Enterprise Miner, a leading commercial data-mining package, was used to create the full video game classification tree shown in Figure 2. The software determines which explanatory variable has the greatest predictive power and then uses a randomly selected hold-back data sample to assess the generalizablity of the split. A split that appears spurious during this validation phase is pruned from the tree.

The strongest explanatory variable in the data set according to Enterprise Miner's splitting heuristic was ReviewScore. The root node of the classification tree in Figure 2 was partitioned into three branches corresponding to games with high, medium, and low reviews from a sample of independent video game critics. This one split provides significant predictive power: If we know nothing else about a new game, we would predict (based on historical data) that it has a 0.34 probability of achieving high sales. However, if we know the new game has earned a high review score (an average of 85 points or greater), the first branch of the predictive model shows that the estimated probability of high sales given a high review score jumps to 0.82 whereas the estimated probability of low sales given a high review score falls 0.03.[2]



Each node shows the distribution of Sales for the training examples represented by the node.

The distribution at each node is conditioned on the path through the classification tree. Thus, $P(\text{Sales = High} \mid x_{10} \geq v_{Thresh}) = 0.9$

Figure 1. A Fragment from a Simple Classification Tree

No other variable in the training data provided a sufficient balance between additional explanatory power and generalizablity to merit a second-level split for either the ReviewScore = high or ReviewScore = low branches in Figure 2. However, the ReviewScore = medium node was split on NumPlatforms (the number of consoles on which the game was released) and a third-level split occurred under Node 5 (NumPlatforms =1) on the ReleaseYear variable. The appearance of ReleaseYear in the predictive model suggests that the determinants of game success have changed over time. Such a split is useful since it helps control for time-based trends by extracting some of the variance in sales due to non-stationary phenomena. However, since the objective here is to predict the performance of future games, only the most recent branch (Node 10: ReleaseYear >= 2004) is relevant to the decision problem.

A fourth-level split occurred under Node 10 on the Quantile(LnPriorSalesBrand) variable, which categorizes first-year sales for all the game's prequels as low, high, or missing. "Missing" is not a data quality error in this case since many games have no prequels and thus have no prior sales for their brand. Classification tree algorithms incorporate missing values by assigning them to their own branch, thereby eliminating the requirement to remove games with missing values from the training data or somehow impute the missing values.

Interpreting the classification tree involves reading the variable = value pairs from the root to a leaf. Thus, traversing from the root to Node 18, we predict that a game with a medium review score that is released for a single platform and which belongs to an established brand of games with strong first-year sales has a 71 percent chance of achieving strong sales and only a 14 percent chance of performing poorly. These estimates provide a valuable gain in information relative to the prior (root node) probability estimates.

Evaluation

CRISP-DM identifies two distinct types of evaluation. The first involves technical measures of the model's predictive accuracy. In classification tasks, a single prediction is assigned to each leaf node based on the most likely value of the response variable at the node. The data-mining tool can estimate the predictive accuracy of the model by: (1) generating predictions for an independent holdback sample of training data; (2) comparing the predicted value to the actual value; and (3) calculating the misclassification rate. However, the practice of assigning a single predicted value to each leaf node effectively wastes the valuable uncertainty information contained in the conditional probability distributions. For example, a leaf node assigned a predicted sales value of "high" based on 34 percent of training examples being "high" (versus 33 percent being "med" and "low") is very different from a leaf node in which 90 percent of the training examples have "high" sales. As we show in Section V, it is both possible and preferable to

[2] The algorithms used to build classification trees may yield biased estimates of class membership probability. Accordingly, the relative frequencies at the leaf nodes should be adjusted using a smoothing transformation before being interpreted as probability estimators [Zadrozny and Elkan 2001; Provost and Domingos 2003]. We omit the smoothing transformations in this discussion in order to simplify exposition.

carry the entire conditional probability distribution forward to the next stage of the decision process. This renders standard measures of predictive accuracy meaningless.

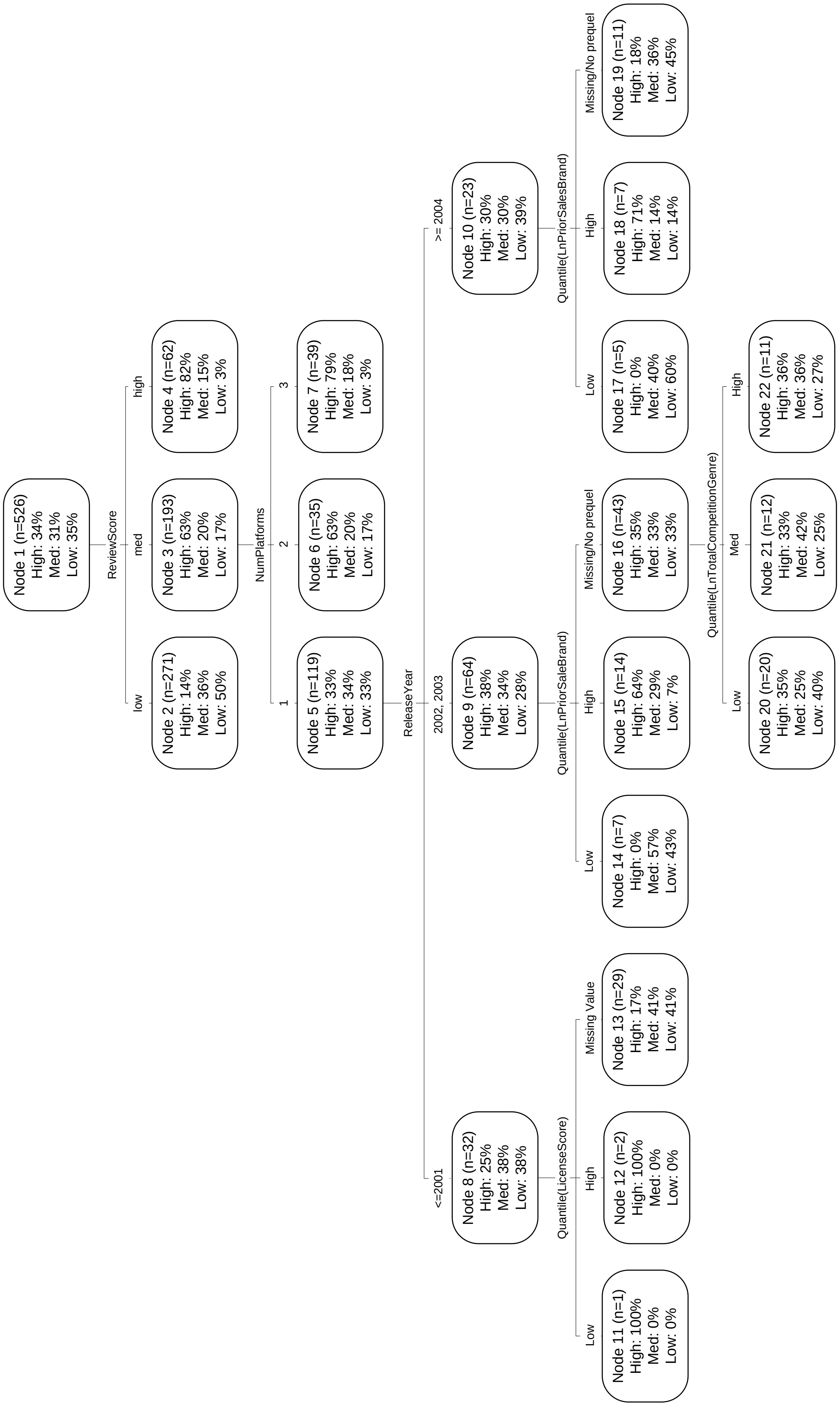


Figure 2. A Classification Tree-Based Predictive Model of Video Game Revenue

The second type of evaluation specified in the CRISP-DM model emphasizes the business meaning of the predictive model: Are the results plausible, novel, and useful? We had to evaluate several elements of the predictive model of video game sales with respect to the underlying business context. For example, we had to confirm the usefulness of the top level split based on ReviewScore given that game reviews typically occur late in a video game's development lifecycle. We determined that ReviewScore was useful, but only as a proxy measure of another unobserved variable. We return to this issue in more detail in Section V.

Deployment

The final "deployment" phase of CRISP-DM emphasizes communication of the results of the data-mining analysis to the client. The implicit assumption is that the client—the decision maker—will be able to use the reports and other outputs from the data-mining analysis as decision making inputs. Thus, according to CRISP-DM, the primary deliverable of a data-mining analysis is not a decision, but rather a report containing a predictive model similar to that shown in Figure 2.

IV. DATA-MINING/DECISION-MAKING INTEGRATION

Given the phases of the CRISP-DM process model described in the previous section, we can now assess the potential role of data mining within each phase of Simon's model of decision making.

Intelligence

The intelligence phase of Simon's model describes information gathering and monitoring activities that alert managers to environmental conditions requiring a decision. In this sense, there is significant correspondence between the initial "business understanding" phase of CRISP-DM and Simon's intelligence phase. However, it is important to recognize that business understanding is defined as *prerequisite* to data mining. According to the CRISP-DM model, no decision support or automation role is defined for the data mining tool during this phase.

If the CRISP-DM process model were extended to include iterations, it is possible that the outcome of one data mining analysis could trigger a new, somewhat tangential, decision problem. For example, in the video game problem, the data-mining results highlight the importance of review scores in predicting game revenues. This might lead development firms to consider introducing better proxies for independent reviews earlier in the game development lifecycle. But despite the potential value of such results, recognizing and exploiting the informational byproducts of previous data-mining analyses is primarily a manual task performed by the decision maker.

Design

The design phase of Simon's model involves the identification or invention of alternatives. One of the difficulties faced by boundedly rational decision makers addressing complex, real-world decisions is an overwhelming number of distinct alternatives . To illustrate, consider the design-phase problem faced by GameCorp . Developers of a new game can control many attributes of the game, such as its genre, target consoles, target market, relationship to licensed brands, and so on. Some of these controllable variables may interact. For example, a game in the children's genre might sell better when targeted to a particular console. Moreover, some controllable variables may be dependent on uncontrollable variables, such as the development decisions of rival development studios. The implication is that game design cannot be viewed as a sequence of independent decisions; instead, developers face a potentially large number of distinct combinations of game configurations.

Data mining can help decision makers cope with a large number of alternatives in two ways. First, the final classification tree contains only the candidate explanatory variables that the algorithm's splitting heuristic deems relevant. Decision makers can use the model to identify and focus on the variables with the largest expected impact on the response variable. For example, the online variable does not appear in the classification tree in Figure 2. This suggests that the decision whether to include online functionality in a new game will have a relatively insignificant impact on the game's expected revenues. Second, the multiple levels of the classification tree permit the identification of dependencies (or the lack thereof) between variables. For example, the classification tree in Figure 2 suggests that a highly rated game has a high probability of doing well regardless of the Genre, ESRB rating, and so on. However, if a game has a middling review score and is released on only one platform, the performance of the game's prequels (if any) becomes a relevant predictor of success.

Challenge 1: Embedded Alternatives

Although classification trees contain valuable information about relevance and interdependence, decision makers cannot simply read a set of alternatives off the tree. The identification of alternatives requires that a distinction be made between variables based on their degree of controllability. The predictive model in Figure 2 provides examples of four distinct types: Controllable variables, such as NumPlatforms, are the deterministic outcomes of decisions. Uncontrollable variables, such TotalCompetitionGenre, are determined by forces over which the decision maker has little or no influence. In between these two extremes are semi-controllable variables and real options.

ReviewScore is an example of a semi-controllable variable since the actions of the decision maker can influence, but not determine, the review scores received by a game. PriorSalesBrand is a special type of semi-controllable variable in which the value of the variable for the current game is influenced by development decisions made in context of past games. We discuss the implications of such "real options" variables in more detail in Section VI. Since the most widely used data-mining algorithms treat all candidate explanatory variables the same regardless of controllability, it is left to the decision maker to extract the alternatives embedded in the predictive model.

Choice

The choice phase of Simon's model consists of two activities: First, the alternatives identified in the design phase are evaluated with respect to a decision criterion such as expected value. Second, the alternative with the highest score is selected for execution. Data mining can have an important role during evaluation because the predictive model provides estimates of an important determinant of value. For example, the leaf nodes of the classification tree in Figure 2 provide estimates of the probability distribution of sales revenues conditioned on specific game attributes. Although such information is valuable during the choice phase, data-mined predictive models typically create two challenges in practice: incomplete evaluation information and uncertainty.

Challenge 2: Incomplete Evaluation Information

In a well-documented data-mining success story, Harrah's used customer survey and transactional information from its casinos to discover that customers who reported being very happy with the Harrah's experience increased their spending on gambling by 24 percent [Loveman 2003]. However, improving customer experience has cost implications: Harrah's spent $14.2 M in 2002 on customer service incentive pay for its employees. The benefit information discovered during data mining must be combined with relevant cost information in order for decision makers to see the whole picture. However, the data sets used for data mining often contain no cost information.

To illustrate the problem in the video game context, consider the ReviewScore variable at the root of the predictive model. GameCorp might reasonably conclude that it should endeavor to create highly rated games since, according to the classification tree, such games have a higher probability of being successful in the market. However, the predictive model was constructed from industry-level data, which reflects a wider array of product development strategies and outcomes than data from a single firm. Although such variation in the data is essential for finding patterns, aggregated data does not contain firm specific, confidential cost information. Accordingly, the predictive model in Figure 2 says nothing about the costs of achieving the benefits it predicts. Additional information from other sources is required before the classification tree can be used for evaluating alternatives.

Challenge 3: Uncertainty

In a perfect predictive model, each leaf node in the classification tree would correspond to a homogeneous subset of data in which all games belonged to the same sales category. In practice, however, the tree's leaf nodes provide discrete probability distributions over outcomes conditioned on the leaf's ancestors. For example, in Node 4 of Figure 2, 82 percent of the games in the data set with ReviewScore = high had high sales; however, the other 18 percent did not.

The residual uncertainty in real-world data mining creates problems when the data mining output is used as an input to deterministic decision-making processes. Currently, the dominant criterion used within firms for evaluating alternatives is net present value (NPV) [Farragher and Kleiman 1999; Ryan and Ryan 2002]. To calculate NPV, firms estimate the benefits of each alternative, subtract the estimated costs, apply a risk-adjusted discount rate to future cash flows, and compare the results to the NPVs of other alternatives. The obvious shortcoming of the NPV approach in the context of data mining is that NPV has no means of dealing with uncertainty or accounting for managerial flexibility in response to uncertain outcomes.

The failure of data-mining methodologies such as CRISP-DM to specify mechanisms for decision making using data-mined models is therefore a serious problem. Although practicing managers may find the reports created during CRISP-DM's deployment phase insightful and valuable, few firms have the capability to incorporate probabilistic predictive models into their formal decision making process.

V. INTEGRATION USING DECISION TREES

According to the analysis in the previous section, data mining can provide valuable but incomplete support to decision makers during the design and choice phases of Simon's model. The purpose of this section is to show how conventional decision analysis tools can be used to bridge the gaps between data mining and decision making.

Decision trees are a well-established means of representing and solving probabilistic decision problems [Raiffa 1968]. Unlike classification trees, decision trees make a clear distinction between chance and decision nodes. It is therefore possible to identify alternatives embedded in the predictive model (Challenge 1) by transforming the classification tree into a decision tree [see Brydon and Gemino, forthcoming] for a more detailed discussion of the transformation). Decision trees also provide a representational formalism that permits probabilistic information from other sources—such as market research or the subjective estimates of experts—to be integrated with the information obtained from data mining. This flexibility enables the benefit estimates from the classification tree to be supplemented with cost information from a variety of other sources (Challenge 2). Finally, the rollback procedure for decision trees provides a principled means of incorporating probabilistic information into the decision-making process (Challenge 3).

To illustrate, recall the challenges posed by the ReviewScore variable in the classification tree in Figure 2. According to the splitting heuristic used by Enterprise Miner, ReviewScore is the single best predictor of sales revenue. However, a game's review score is clearly not a development decision; rather, it is an outcome of other development decisions about which we have no information. According to our contacts at GameCorp, the best controllable predictor of ReviewScore is development effort (which includes such factors as the quality of the graphics, the details of the game scenarios, and so on). Since we had no information about development effort in our data set, we created a new controllable variable called DevEffort and asked a panel of seven producers and developers at GameCorp to estimate the relationship between the DevEffort and ReviewScore. The panel was first asked to categorize development cost by providing dollar estimates of high, medium, and low development effort ($17.14M, $7.79M, and $3.86M respectively). Panel members were then asked to review a sample of 265 video games and estimate the development effort that would have been required for GameCorp to create each game. Naturally, such estimates are subject to uncertainty: First, panel members are not able to perfectly estimate the development cost of games developed by other firms. Second, although a positive statistical correlation exists between development effort and review score, there is no guarantee that a game that is costly to develop will receive critical acclaim.

Decision trees provide a natural means of distinguishing alternatives from outcomes and combining the probabilistic information from the experts with the probabilistic information created by data-mining software. Consider the decision tree for the video game develop project shown in Figure 3. The initial decision node corresponds to development effort. The top two branches for high and medium DevEffort have the same structure as the branch for low-development effort and have been collapsed to save space. The panel's cost estimates for the different levels of development effort are shown along the decision arcs connecting the "DevEffort" decision nodes with the "Review" chance nodes. The conditional probabilities relating review scores to sales performance—which were provided by the classification tree and smoothed in accordance with the procedure described by [Zadrozny and Elkan 2001]—are shown along the chance arcs emanating from the ReviewScore nodes. Finally, the expected revenues for high, medium, and low sales are shown below the probabilities.

Once a decision tree is fully parameterized with cost, probability, and payoff information, it is a simple matter to rollback the tree to determine the value of the alternatives at the root decision node. The commercial software used to create the decision tree in Figure 3 has designated the optimal decision arcs with "True" and all others as "False." According to this decision tree and the firm-specific data it contains, GameCorp should invest less in development and accept modest sales (expected net benefit: $8.06M) rather than invest heavily in development in the hope of producing a blockbuster (expected net benefit: $1.58M).[3]

The firm's development strategy can be further refined by incorporating additional information from the next level from the classification tree. Recall from the predictive model that the number of platforms on which a game is released is a predictor of total revenue in certain circumstances. However, the net benefit of developing a game for multiple consoles depends on firm-specific and game-specific porting costs. The decision tree analysis indicates that GameCorp should be willing to pay as much as $3.7M in additional development costs to release the game on two consoles and up to $6M to release the game on all three consoles if the game receives middling reviews.

VI. DEALING WITH REAL OPTIONS

As the results in the previous section illustrate, decision trees provide a means of incorporating a probabilistic predictive model from data mining into a full probabilistic decision model. However, there are practical limits to the use of decisions trees for data-mining/decision-making integration that emerge when the decision problem contains real options. To illustrate, consider the addition of the Quantile (LnPriorSalesBrand) variable to the decision tree. Like ReviewScore, the revenue earned by previous games in the same brand family is a semi-controllable outcome of one or more development decisions. However, unlike ReviewScore, Quantile (LnPriorSalesBrand) is an outcome

[3] The results in this section are based on firm-specific cost information which has been disguised. Our objective here is to illustrate a methodology rather than to provide specific guidance to video game developers.

of decisions made for *previous* games. A decision tree for a single game can therefore systematically underestimate the value of a blockbuster outcome since it fails to account for the possibility of a successful game leading to successful sequels in the future.

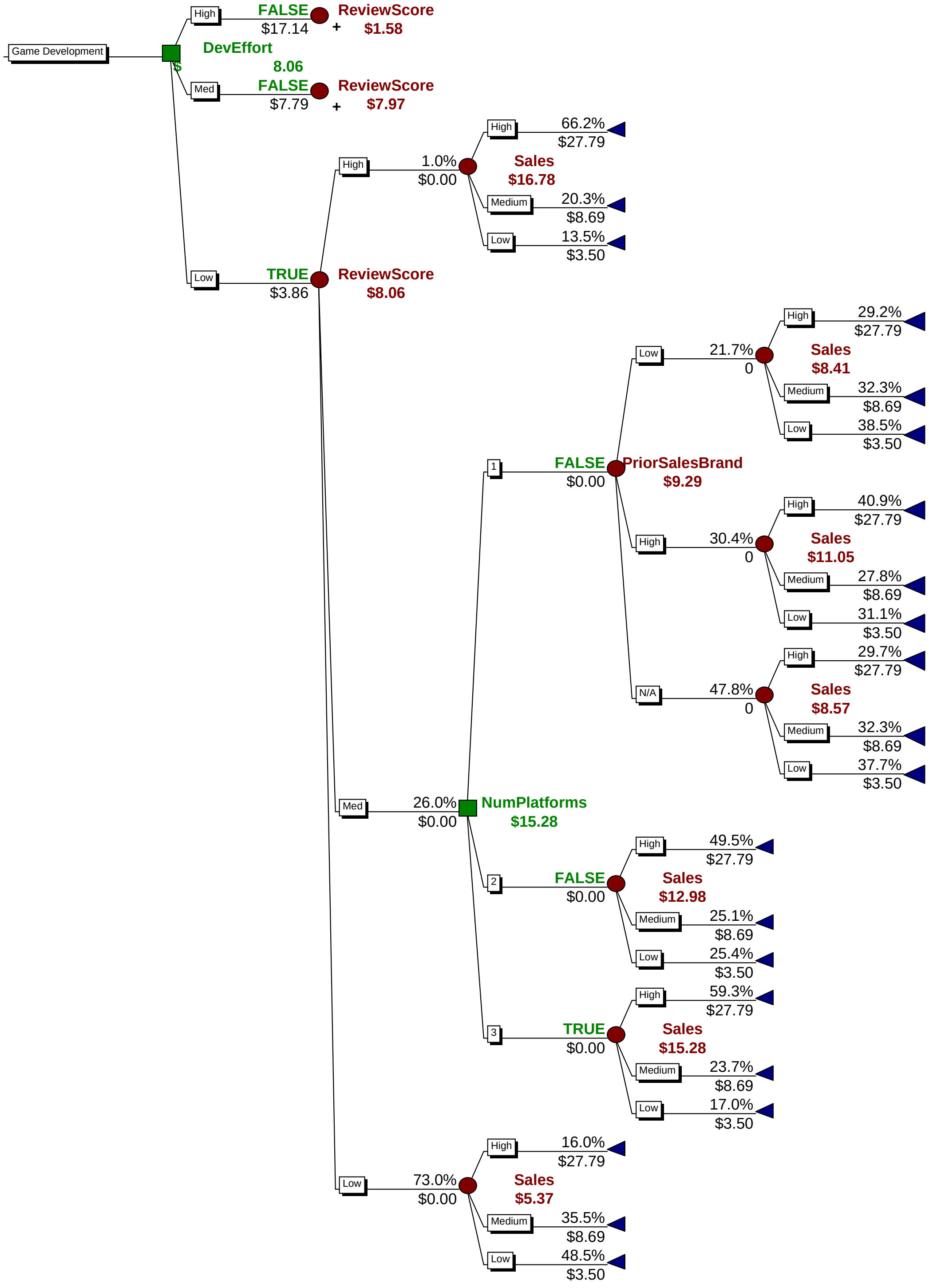


Figure 3. A Decision Tree Fragment Showing the Relationship between Development Effort, Review Score, and Total Revenue

A sequence in which a decision is made, uncertainty is resolved, and a second decision is made corresponds to a real option [see Trigeorgis 1996]. In this context, investments in a game give the development firm the option to observe the game's performance and, if appropriate, develop a sequel. Option-like structures in a predictive model greatly complicate the use of decision trees since the evaluation of alternatives for a game requires the simultaneous evaluation of alternatives for a portfolio of decision problems. To illustrate, consider the simplified decision tree in Figure 4, which consists of a single binary decision (whether to develop a game of a particular brand) and a single binary chance outcome (the revenue earned by the brand). Due to the interdependence created by prior sales, the decision trees for the two games must be chained together. Even under this simple formulation, the combined decision tree for a two-game formulation grows very quickly. If we assume that the decision tree for a single game has 20 leaf nodes (a relatively modest number), then the full model for a portfolio of five games requires a tree with roughly 20^5 or 3.2M leaf nodes.

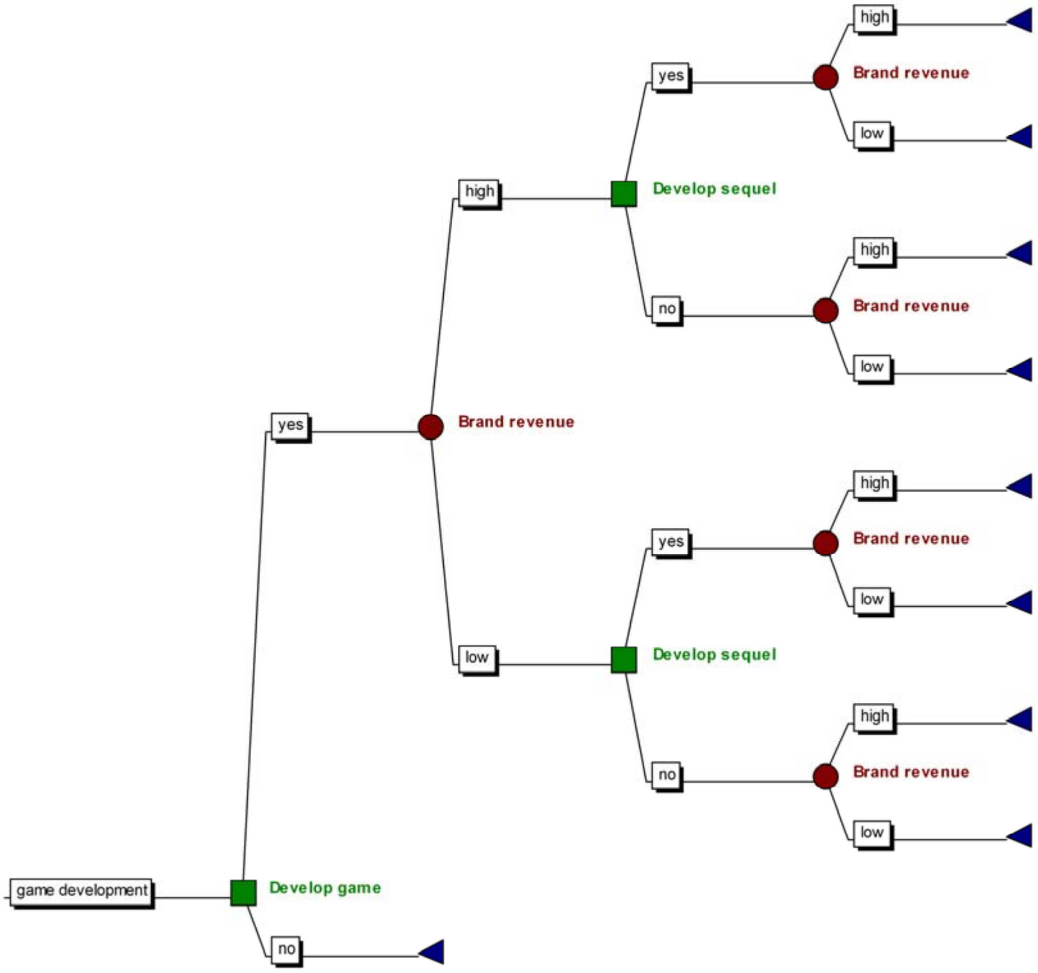


Figure 4. A Simplified Decision Tree for the Sequel Option

This combinatorial explosion in problem size renders decision trees impractical for virtually any real-world decision problem containing real options. More importantly, the computational challenges posed by real options are not idiosyncratic to decision trees. Real options create a tight linkage between the valuation problem and a broader, more difficult stochastic planning problem.

Several techniques have been proposed for estimating the value of real options without explicitly evaluating of millions of contingencies. Some techniques are based on a direct application of option pricing models from finance [Trigeorgis 1996; Luehrman 1998; Copeland and Tufano 2004]. However, the financial-option-pricing approach suffers from two well-known shortcomings: First, decision makers often have difficulty understanding the theoretical underpinnings of analytical-pricing techniques such as the Black-Scholes and binomial lattice models [Lander and Pinches 1998]. Second, the parameters required by the models are readily available for financial options but are often difficult to estimate in decision contexts in which complete markets are nonexistent.

A possible alternative to the opaqueness and inflexibility of financial option pricing models and the impracticality of large decision trees is *decision-theoretic planning*. Decision-theoretic planning describes techniques developed within the AI planning community for controlling robots in complex and uncertain environments [Boutilier et al. 1999]. As the name suggests, the foundation of decision-theoretic planning is classical decision theory—in particular, the formulation of planning as a Markov decision problem (MDP) and the use of dynamic programming to assign optimal actions to various states of the world [Puterman 1994].

The primary difference between decision-theoretic planning and decision-tree analysis is that a decision tree cannot be solved until it is built-out for every possible course of action, including those that are inferior. In a state-based approach such as decision-theoretic planning, a dynamic programming algorithm makes provisional commitments to actions in every possible system state and refines the choices through multiple iterations. This eliminates the requirement to construct a massive tree and reduces the burden placed on the decision analyst.

The primary contribution of the AI community to the MDP/dynamic programming foundation has been the development of highly structured and parsimonious languages for problem representation. These structured problem representations can be used by specialized dynamic programming algorithms to determine an optimal policy without fully enumerating the problem's state space. The result is often a significant decrease in the effective size of the planning problem [Brydon 2006].

Although a detailed discussion of decision-theoretic planning is beyond the scope of this paper, we have applied decision theoretic planning to a development portfolio of five video games. Our results suggest that decision-theoretic planning can indeed be used to solve problems of realistic size and complexity, even if the problems contain real options. Although manipulation of the structured problem representations entails significant computational overhead, the effective size of the final optimal "policy tree" in our analysis was a modest: 1,730 nodes. This compares favorably to the 3.2M nodes that would be required for a complete decision-tree analysis of the same problem. Interestingly, even when the option value of sequels over a five-game horizon is accounted for, the cost and risk of a blockbuster strategy for GameCorp remains too high. The best policy for the firm is to allocate its finite development resources to several low-cost games rather than invest heavily in a single potential blockbuster.

VII. CONCLUSIONS

The emergence of mature commercial data-mining packages and the increased availability of large amounts of data have put sophisticated predictive modeling within the reach of many firms. Accurate predictive models can be extremely useful in industries in which decision makers must make one-shot, irreversible decisions early in their product development lifecycles. However, data mining is only the first step toward realizing a payoff from investments in data and data analysis. Using data-mining output as an input into a firm's decision-making processes remains challenging regardless of whether the predictive models are created using off-the-shelf software or the latest, most advanced data-mining algorithms.

Our objective in this paper was to use a real-world example to highlight the gap between the sophistication and power of data-mining software and the techniques conventionally used within firms to make decisions. Well-established decision analysis tools such as decision trees provide a partial solution to the data-mining/decision-making integration problem. Emerging techniques, such as decision theoretic planning, extend the boundary of what can be solved in practice just far enough to permit evaluation of problems containing simple real options.

Our view, given the uncertainty and high stakes in industries such as video game development, is that investments in data and data mining will enable firms to make better decisions. However, it is important that firms recognize that data mining is only one element in a larger decision making process. Tools and techniques for data mining/decision making integration are still in their infancy and have yet to approach the sophistication, ease of use, and widespread adoption of commercial data mining software. Firms must be willing to reconsider the ways in which they make decisions if they are to realize a payoff from their investments in data mining technology.

REFERENCES

Alter, S. (2004). "A Work System View of DSS in Its Fourth Decade," *Decision Support Systems 38*(3), 319-327.

Apte, C., L. Bing, E. P. D. Pednault, and P. Smyth. (2002). "Business Applications of Data Mining," *Communications of the ACM 45*(8), 49-53.

Arnott, D. (2006). "Cognitive Biases and Decision Support Systems Development: A Design Science Approach," *Information Systems Journal 16*(1), 55-78.

Boutilier, C. et al. (1999). "Decision-Theoretic Planning: Structural Assumptions and Computational Leverage," *Journal of Artificial Intelligence Research* 11, 1-94.

Box, George E. P. (1957). "Evolutionary Operation: A Method for Increasing Industrial Productivity," *Applied Statistics 6*(2), 81-101.

Brooks, R. A. (1991). “Intelligence without Reason,” *Proceedings of the 12th International Joint Conference on Artificial Intelligence,* 569-595.

Brydon, M., (2006) “Evaluating Strategic Options Using Decision-Theoretic Planning,” *Information Technology and Management* 7(1) 35-49.

Brydon, M., and A. Gemino. (Forthcoming). “Classification Trees and Decision-Analytic Feed Forward Control: A Case Study from the Video Game Industry,” Forthcoming in *Data Mining and Knowledge Discovery.*

Chapman, P. et al. (2000). *CRISP-DM 1.0: A Step-By-Step Data Mining Guide*, http://www.crisp-dm.org/Process/index.htm , accessed May 5, 2007.

Copeland, T., and P. Tufano. (2004). “A Real-World Way to Manage Real Options,” *Harvard Business Review* 82(3), 90-99.

Davenport, T. H. (2006). “Competing on Analytics,” *Harvard Business Review 84*(5), 150-151.

Davenport, T. H., et al. (2001). “Data to Knowledge to Results: Building an Analytic Capability,” *California Management Review 43*(2), 117-138.

Einhort, H. J., and R. M. Hogarth. (1981). “ Behavioral Decision Theory: Processes of Judgment and Choice ,” *Journal of Accounting Research 19*(1), 1-31.

Farragher, E. J., and R. T. Kleiman. (1999). “ Current Capital Investment Practices ,” *Engineering Economist 44*(2), 137.

Gaume, N. (2006) “Nicolas Gaume’s Views on the Video Games Sector,” *European Management Journal* 24(4), 299-309.

Hannula, M., and V. Pirttimaki. (2003). “Business Intelligence Empirical Study on the Top 50 Finnish Companies,” *Journal of American Academy of Business 2*(2), 593-601.

Haughton, D. et al. (2003). “A Review of Software Packages for Data Mining,” *American Statistician 57*(4), 290.

Jackson, J. (2002). “Data Mining: A Conceptual Overview,” *Communications of the Association for Information Systems 8*, 267-296.

Khatri, N., and H. A. Ng. (2000). “The Role of Intuition in Strategic Decision Making,” *Human Relations 53*(1), 57-86.

Kohavi, R., N. J. Rothleder, and E. Simoudis. (2002). “Emerging Trends in Business Analytics,” *Communications of the ACM 45*(8), 45-48.

Lander, D. M. and G. E. Pinches. (1998). “Challenges to the Practical Implementation of Modeling and Valuing Real Options,” *Quarterly Review of Economics and Finance* 38(4) 537.

Loveman, G. (2003). “Diamonds in the Data Mine,” *Harvard Business Review 81*(5), 109-113.

Luehrman, T. A. (1998). “Strategy as a Portfolio of Real Options,” *Harvard Business Review 76*(5), 89-99.

Negash, S. (2004). “Business Intelligence,” *Communications of AIS* 2004(13), 177-195.

Provost, F., and P. Domingos. (2003). “Tree Induction for Probability-Based Ranking,” *Machine Learning 52*(3), 199-215.

Puterman, M. L. (1994). *Markov Decision Processes :Discrete Stochastic Dynamic Programming*, John Wiley & Sons, New York.

Raiffa, H. (1968). *Decision Analysis: Introductory Lectures on Choices under Uncertainty*. Reading, MA: Addison-Wesley.

Reimer, J. (2005, November 07, 2005). *Cross-Platform Game Development and the Next Generation of Consoles*. [Electronic version]. *Ars Technica,*

Ryan, P. A., and G. P. Ryan. (2002). “Capital Budgeting Practices of the Fortune 1000: How Have Things Changed?” *Journal of Business and Management 8*(4), 355.

Simon, H. A. (1977). *The New Science of Management Decision* (Rev. -- ed.). Englewood Cliffs, N.J.: Prentice-Hall.

Simon, H. A. (1955). “A Behavioral Model of Rational Choice,” *Quarterly Journal of Economics 69*(1), 99-118.

Smith, J. E. (2005). “Alternative Approaches for Solving Real-Options Problems,” *Decision Analysis 2*(2), 89-102.

Trigeorgis, L. (1996). *Real Options : Managerial Flexibility and Strategy in Resource Allocation*. Cambridge, MA: MOT Press.

Zadrozny, B., and C. Elkan. (2001). "Obtaining Calibrated Probability Estimates from Decision Trees and Naive Bayesian Classifiers," *Proc. 18th International Conf. on Machine Learning,* 609-616.

ABOUT THE AUTHORS

Michael Brydon is an Assistant Professor in the Faculty of Business Administration at Simon Fraser University in Vancouver. He received his Ph.D. in Management Information Systems from the University of British Columbia and M.Eng. and B.Eng. degrees in Engineering Management from the Royal Military College of Canada. His research interests lie at the intersection of decision theory, economics, and computer science and include computational economies, decision-theoretic valuation of real options, and markets for public goods such as knowledge and open source software. Recent articles have appeared in *Decision Support Systems, Information and Technology Management* and *Data Mining and Knowledge Discovery.*

Andrew Gemino is an associate professor in the Faculty of Business Administration at Simon Fraser University. His research interests include information technology project management, business analysis and decision making. His work has been published in *JMIS, Communications of the ACM, Data Mining and Knowledge Discovery* and the *European Journal of IS*. His research on IT project management can be found at www.PMPerspectives.org. He is funded through grants from the National Sciences and Research Council (NSERC) and the Social Sciences and Humanities Research Council (SSHRC). Andrew recently co-authored a textbook entitled *Experiencing MIS* for Pearsons Education Canada. He is President of the AIS Special Interest Group on Systems Analysis and Design (SIGSAND) and also volunteers for the Surgeon Information System Working Group associated with the BC Cancer Agency.

Communications of the Association for Information Systems

ISSN: 1529-3181