A Decision-Theoretic Approach to Data Mining

Yuval Elovici and Dan Braha

***Abstract*—In this paper, we develop a decision-theoretic framework for evaluating data mining systems, which employ classification methods, in terms of their utility in decision-making. The decision-theoretic model provides an economic perspective on the value of "extracted knowledge," in terms of its payoff to the organization, and suggests a wide range of decision problems that arise from this point of view. The relation between the *quality* of a data mining system and the amount of investment that the decision maker is willing to make is formalized. We propose two ways by which independent data mining systems can be combined and show that the combined data mining system can be used in the decision-making process of the organization to increase payoff. Examples are provided to illustrate the various concepts, and several ways by which the proposed framework can be extended are discussed.**

***Index Terms*—Actionability, classification, data mining, data mining economics, decision-making, knowledge discovery systems.**

I. Introduction

POWERFUL data acquisition systems (such as minicomputers, microprocessors, transducers, and analog-to-digital converters) that collect, analyze, and transfer data are in use in various mid-range and large organizations [2], [4]–[7], [27]. Over time, more and more current, detailed, and accurate data are accumulated and stored in databases at various stages. This data may be related to designs, products, machines, materials, processes, inventories, sales, marketing, and performance data and may include patterns, trends, associations, and dependencies. The data collected contain valuable information that could be integrated within the organization strategy, and used to improve organization decisions.

The large amount of data in current databases, which contain large number of records and attributes that need to be simultaneously explored, makes it almost impractical to manually analyze them for valuable decision-making information. The need for automated analysis and discovery tools for extracting useful knowledge from huge amounts of raw data suggests that knowledge discovery in databases (KDDs) and data mining methodologies may become extremely important tools in realizing the above objectives. Some researchers often define data mining as the process of extracting valid, previously unknown, comprehensible information from large databases in order to improve and optimize organization decisions [5], [23]. Other researchers use the term KDD to denote the entire process of turning low-level data into high-level knowledge, where data mining is considered as a single step in the process that involves finding patterns in the data. To avoid confusion, we choose the later definition.

The KDD process is defined in [5] as "the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data." According to [2], although data mining is at the core of the KDD process, it is just one step in the overall KDD process, and it usually takes about 15 to 25% of the overall effort. The KDD process often includes the following important stages [5]. The first step involves understanding the application domain in which the data mining is applied and the goals of the data mining process. The second step includes selecting, integrating, and checking the target data set. The target data set may be defined in terms of the records as well as the attributes of interest to the decision-maker. The third step is data preprocessing. This includes data transformation, handling missing or unknown values, and data cleaning (this can be done by applying algorithms in order to remove unreliable and erroneous data). In the fourth step, data mining for extracting patterns from data takes place. This involves model and hypothesis development and the selection of appropriate data mining algorithms. The fifth step involves interpreting and presenting results for the decision-maker.

Fayyad *et al.* [5] distinguish between two main categories of data mining (fourth step above): *verification-oriented* and *discovery-oriented*. Verification-oriented techniques focus mainly on testing preconceived hypotheses (generated by the decision-maker) and on fitting models to data. Discovery-oriented methods focus on autonomously finding new rules and patterns and are classified as *descriptive* or *predictive*. Descriptive methods include *visualization* techniques (e.g., scatter plots and histograms) and *clustering* (e.g., identifying subgroups of silicon wafers that have a similar yield). Predictive methods include *regression* and *classification*. Regression is concerned with the analysis of the relationships between attribute values within the same record and the automatic production of a model that can predict attribute values for future records. Classification methods assign records to predetermined classes. For example, medical patients may be classified according to the outcome of their treatment; thus, the most effective treatments for new patients can be identified.

In this paper, our focus is on data mining systems that employ classification methods. While much research has been conducted on making optimal cost-sensitive classification decisions [24], there is virtually no rigorous and formal research related to the question of *actionability*—the ability of the "extracted knowledge" to suggest concrete and profitable action by the decision-makers [8], [9], [14], [16]. The difficulty of determining the value of "mined data" and the tangible benefits resulting

Manuscript received March 13, 2000; revised December 22, 2002. This paper was recommended by Associate Editor G. Biswas.

Y. Elovici is with the Department of Information Systems Engineering, Ben-Gurion University, Beer-Sheva 84105, Israel.

D. Braha is with the School of Industrial Engineering and Management, Ben-Gurion University, Beer-Sheva 84105, Israel (e-mail: braha@mit.edu).

Digital Object Identifier 10.1109/TSMCA.2003.812596

from investing for an organization to investment in the KDD process keeps many organizations from fully exploiting the affluence of data that is generated and collected during daily operations. The importance of this question increases even more when considering that the market for data mining has grown from $50 million in 1996 to $800 million in 2000 [7]. Moreover, many organizations use data mining as a strategic tool in order to become more competitive.

The purpose of this paper is to develop a framework for evaluating data mining systems, which use classification methods, in terms of their value in decision-making. Our framework is based on the belief that the question of evaluating data mining systems can only be addressed in a *utilitarian* framework, that is, the patterns extracted by the data mining system are effective only to the extent that the derived information leads to action that increases the *payoff* of the decision-maker (see [8] and [18] for a similar view). The decision-theoretic framework developed in this paper connects the organization's strategic objectives with KDD investment and data mining quality. This helps in understanding how KDD benefits change as a function of the deployment cost of the KDD process, what should be the optimal investment in KDD, and what is the nature of the relationship between the organizational strategy and data mining quality. Our modeling approach also enables us to address the question of evaluating different data mining processes when making decisions.

The paper is organized as follows. Section II presents the basic decision-theoretic framework and introduces an ensemble method that combines into one composite data mining system two or more data mining systems. Section III relates the effectiveness of the KDD process to its cost. Section IV extends the ensemble method presented in Section II. Section V concludes the paper. The proofs of the theorems are presented in [30].

II. Decision–Theoretic Classification Model

A. Basic Notation and Definitions

As mentioned above, at the core of the KDD process are the data mining methods for extracting patterns from data. These methods can have different goals, depending on the intended outcome of the overall KDD process [18]. In this paper we analyze KDD processes, which employ data mining methods that fall under the category of classification (e.g., neural networks, Bayesian networks, decision trees, and example-based learning [12]).

Our modeling approach is based on the information structure model presented by Marschak [11], McGuire and Radner [10], Demski [3], and others who expanded the model [1]. The proposed decision-theoretic framework to data mining in its simplest form is as follows. Certain *examples* (instances) are to be labeled as coming from a set of *actual classes* S, where $S = \{s_1, \ldots, s_{n_S}\}$. Let $\pi = (\pi_1, \ldots, \pi_{n_S})$ denote the vector of prior probabilities of the actual classes in the population under study. Each example gives rise to certain attributes, which together form the attribute vector X. The task of the data mining system is to classify an example to one of n_y *predicted classes* in $Y = \{y_1, \ldots, y_{n_y}\}$ on the basis of the observed value $X = x$. We use the labels $Y = \{y_1, \ldots, y_{n_y}\}$ for the classifications produced by a model, in order to distinguish between the actual class and the predicted class of an example. In most classification systems, the set of actual classes S is identical to the set of predicted classes Y; however, we need the extended definition as will be explained later. The data mining measure of performance is defined in terms of the *confusion matrix* P [22]. The confusion matrix is a stochastic (Markovian) matrix of size $n_s \times n_y$ of conditional probabilities, where each of its elements p_{ij} defines the probability of deciding a predicted class y_j given an example of actual class s_i.

The decision-maker observes predicted classes as determined by the data mining system and chooses actions accordingly (e.g., "make sales promotion offers by direct mail"). This is fundamentally different from the classical formulation of the cost-sensitive classification problem,[1] where the sets of classes and actions are assumed to be identical. Let $A = \{a_1, \ldots, a_{n_a}\}$ be a finite set of actions that can be taken by the decision-maker and U a cardinal payoff matrix of size $n_a \times n_s$ that associates payoffs with pairs of actions and actual classes, i.e., $u_{i,j}$ denotes the payoff when the decision-maker applies action a_i, and the example actual class turns out to be s_j.

The decision rule is described by a stochastic (i.e., Markovian) matrix D of size $n_y \times n_a$, where each element $d_{i,j}$ denotes the probability that the decision-maker applies action a_j given the predicted class y_i.

The expected payoff for a decision rule D is given by

$$EU = \text{trace}(PDU\Pi) \tag{1}$$

where the square matrix Π is obtained by placing the vector of prior probabilities π in the main diagonal and zeros elsewhere, and the trace operator denotes the sum of the elements of the main diagonal.

The decision-maker wishes to *maximize* the expected payoff as given by (1). This is achieved by choosing an optimal decision matrix $D^* \in \mathcal{D}$, where $\mathcal{D}$ is the set of all Markovian matrices. Since the expected payoff as given by (1) is linear in the elements $d_{i,j}$ of the decision matrix D, the optimal decision rule can be obtained by solving the following *linear programming problem*:

$$\max_{d_{ij}} \quad \text{trace}(PDU\Pi)$$

subject to

$$\begin{aligned} &\sum_{j=1}^{n_a} d_{ij} = 1, && \text{for } i = 1, 2, \ldots, n_y \\ &d_{ij} \geq 0, && \text{for } i = 1, 2, \ldots, n_y,\ j = 1, 2, \ldots, n_a. \end{aligned} \tag{2}$$

The constraints in (2) follow from the properties of a stochastic matrix. It can be shown that at least one of the optimal solutions is in the form of a decision matrix whose elements are 0 or 1 (a *pure* decision rule). This fact can be exploited to obtain an efficient algorithm for producing the optimal decision rule (see

[1] The common approach employs Bayes optimal prediction, which assigns each example to the class that minimizes the conditional risk (that is, the expected cost of predicting that an example belongs to a particular class). The Bayes optimal prediction is guaranteed to achieve the lowest possible overall expected cost (e.g., [25]).

[26]). Notice, that if the decision-maker adopts a pure decision rule, then the stochastic matrix becomes an "indicator matrix."

B. Comparing Data Mining Systems

Given two data mining systems that are used to classify examples coming from the *same* set of actual classes, we can contrast their measures of performance by comparing the corresponding confusion matrices:

Definition 1: The confusion matrix Q is *more effective* than the confusion matrix R if the maximal expected payoff yielded by R is *not larger* than that yielded by Q for **all** payoff matrices U and **all** prior probability matrices Π.

A partial rank ordering of confusion matrices is provided by the following theorem [10]:

Theorem 1: The confusion matrix Q is *more effective* than the confusion matrix R if and only if there exists a stochastic matrix M with appropriate dimensions such that $Q \cdot M = R$.

Example 1: Consider two classifiers represented by the following confusion matrices:

$$P = \begin{bmatrix} 0.6, & 0.2 & 0.2 \\ 0.2, & 0.6 & 0.2 \\ 0.2, & 0.2 & 0.6 \end{bmatrix}, \qquad Q = \begin{bmatrix} 0.9, & 0.1 & 0 \\ 0.1, & 0.9 & 0 \\ 0.25, & 0.25 & 0.5 \end{bmatrix}.$$

It can be checked that the Markovian matrix

$$M = \begin{bmatrix} 0.65, & 0.15 & 0.2 \\ 0.15, & 0.65 & 0.2 \\ 0, & 0 & 1 \end{bmatrix}$$

satisfies $Q \cdot M = P$; thus, according to Theorem 1, the classifier represented by Q is *more effective* than the classifier represented by P, *regardless of payoff or class distribution information*.

Example 2: The following simple example[2] is provided to illustrate the above notation. Consider a loan screening application in which applicants for a loan from a bank are classified as one of three classes: "low," "medium," or "high" payment risks. The bank has applied a data mining algorithm to its database of previous loan applications and their payment results and has induced a classifier that is used to classify future loan applications. The set of actions that can be taken by the bank (the decision-maker), based on the currently employed classifier's prediction, includes "approve application" or "reject an application." The consequence of rejecting a low payment risk applicant carries a certain reputation cost; the cost of approving a loan for a high payment risk applicant can be much higher. The above information is formalized as follows:

- *Actual Classes* (S) of an applicant: {"low risk," "medium risk," "high risk"};
- *Prior Probabilities* $\{\pi\}$: (0.8 for "low risk," 0.15 for "medium risk," 0.05 for "high risk"};
- *Predicted Classes* (Y): {"low risk," "medium risk," "high risk"};
- *Actions* (A): {"approve application," "reject application"}
- *Payoff Matrix*

$$U = \begin{array}{cccc} & \multicolumn{3}{c}{\textbf{Actual Class}} \\ \textbf{Action} & \text{low} & \text{medium} & \text{high} \\ \text{approve} & 200 & 100 & -1000 \\ \text{reject} & -30 & -20 & 20 \end{array}$$

- *Confusion Matrix*

$$P = \begin{array}{cccc} & \multicolumn{3}{c}{\textbf{Predicted Class}} \\ \textbf{Actual Class} & \text{low} & \text{medium} & \text{high} \\ \text{low} & 0.7 & 0.2 & 0.1 \\ \text{medium} & 0.1 & 0.8 & 0.1 \\ \text{high} & 0.05 & 0.15 & 0.8 \end{array}$$

The optimal decision rule D, which is obtained by maximizing the expected payoff via solving the linear programming problem as given by (2), is the following.

- *Optimal Decision Rule*

$$D = \begin{array}{ccc} & \multicolumn{2}{c}{\textbf{Action}} \\ \textbf{Predicted Class} & \text{approve} & \text{reject} \\ \text{low} & 1 & 0 \\ \text{medium} & 1 & 0 \\ \text{high} & 0 & 1 \end{array}$$

By plugging this optimal decision rule in the expected payoff EU as given by (1), we obtain the maximum expected payoff of 145.6.

Let us assume that the bank is considering modifying the existing data mining system by employing a new classifier for which the confusion matrix is

$$Q = \begin{array}{cccc} & \multicolumn{3}{c}{\textbf{Predicted Class}} \\ \textbf{Actual Class} & \text{low} & \text{medium} & \text{high} \\ \text{low} & 0.9 & 0.05 & 0.05 \\ \text{medium} & 0.1 & 0.8 & 0.1 \\ \text{high} & 0.1 & 0.15 & 0.75 \end{array}$$

The maximum expected payoff that can be achieved by this classifier is: 152.25. Thus, the new classifier represented by the confusion matrix Q is preferred to the existing classification system.[3]

C. Cartesian Composite Classification

Next, we show how several *independent* data mining processes that are used to classify examples coming from the *same* set of actual classes can be combined into one data mining process. To illustrate, we might use a neural network classifier and a decision tree classifier, which is used to classify examples coming from the *same* set of actual credit risk classes: *low risk, medium risk, or high risk*. The outputs (i.e., predicted classes) of these classifiers are then combined (as discussed below), and the decision-maker chooses whether to approve an applicant or not based on the composite classification. Data mining systems

[2] Even though the examples presented in this paper are related to business retail, obviously the approach introduced in this paper would be useful in other domains as well (e.g., medical or manufacturing applications). For instance, in [26] the decision-theoretic framework is applied to a real-world problem of production control in the semiconductor industry.

[3] This does not imply, of course, that Q is preferred to P for *all* payoff and class distribution information (i.e., the case where Q is *more effective* than P).

are probabilistically independent if the probability of deciding by one data mining system that an example of actual class s_i belongs to class y_j does not depend on the classification produced by the other data mining system.

First, we define the *Cartesian product* of two vectors v and w by taking all possible products between the elements of v and those of w. For example, if v is a 1 by n vector and w is a 1 by m vector, then $v \times w$ is the 1 by $n \cdot m$ vector

$$v \times w = (v_1 \cdot w_1, v_1 \cdot w_2, \ldots, v_1 \cdot w_m, \ldots \\ v_n \cdot w_1, v_n \cdot w_2, \ldots, v_n \cdot w_m).$$

The *Cartesian* product of two matrices A and B of size $n \times m_1$ and $n \times m_2$, respectively, is defined as follows:

$$A \times B = \begin{pmatrix} a_1 \times b_1 \\ a_2 \times b_2 \\ \vdots \\ a_n \times b_n \end{pmatrix}$$

where a_i (respectively, b_i) is the ith row in A (respectively B).

Consider two independent data mining processes that are used to classify examples coming from the *same* set of actual classes (i.e., $n_s^P = n_s^Q = n_s$). Let P and Q denote the associated confusion matrices of size $n_s \times n_y^P$ and $n_s \times n_y^Q$, respectively. The two independent data mining processes can be combined into one data mining process called the *Cartesian process*, whose confusion matrix R of n_s rows and $n_y^P \cdot n_y^Q$ columns is defined as follows (for a similar definition see [1])

$$R \quad = P \times Q.$$

The Cartesian process is used to classify examples coming from the *same* set of actual classes as that of P and Q, and it classifies an example to one of the predicted classes in the set $Y^P \times Y^Q$.[4]

Example 3: Consider two independent classifiers represented by the following confusion matrices:

$$P = \begin{array}{c} \\ s_1 \\ s_2 \end{array}\!\!\begin{array}{c} \begin{array}{cc} y_1^P & y_2^P \end{array} \\ \begin{pmatrix} 0.8, & 0.2 \\ 0.3, & 0.7 \end{pmatrix} \end{array}, \qquad Q = \begin{array}{c} \\ s_1 \\ s_2 \end{array}\!\!\begin{array}{c} \begin{array}{cc} y_1^Q & y_2^Q \end{array} \\ \begin{pmatrix} 0.6, & 0.4 \\ 0.2, & 0.8 \end{pmatrix} \end{array}.$$

The Cartesian composite classifier is used to classify examples coming from the set of actual classes $S^{\text{Cartesian}} = \{s_1, s_2\}$. The task of the classifier is to classify an example to one of the *four* classes in $Y^{\text{Cartesian}} = \left\{y_1^P \,\&\, y_1^Q, y_1^P \,\&\, y_2^Q, y_2^P \,\&\, y_1^Q, y_2^P \,\&\, y_2^Q\right\}$ on the basis of the observed value $X = x$. The confusion matrix associated with the Cartesian composite classifier is:

$$R = P \times Q$$

$$= \begin{array}{c} \\ s_1 \\ s_2 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} y_1^P \,\&\, y_1^Q & y_1^P \,\&\, y_2^Q & y_2^P \,\&\, y_1^Q & y_2^P \,\&\, y_2^Q \end{array} \\ \begin{pmatrix} 0.48 & 0.32 & 0.12 & 0.08 \\ 0.06 & 0.24 & 0.14 & 0.56 \end{pmatrix} \end{array}.$$

The Cartesian composite classifier is implemented as follows. First, an example (say with actual class s_2) is classified by both classifiers, and their predictions are concatenated to form one of the four predicted classes in $Y^{\text{Cartesian}}$. The decision-maker observes the predicted class (say $y_1^P \,\&\, y_2^Q$) and chooses the *optimal action* (say a_1) accordingly from within the set of available actions $A^{\text{Cartesian}} = \{a_1, a_2\}$. In other words, both classifiers are run, and a payoff $u_{1,2}$ is associated with the action a_1 and actual class s_2.

It can be shown (see [1]) that the Cartesian process produces a "more effective" classifier (see Definition 1) that is at least as good (in terms of maximization of payoffs) as any of the component classifiers *for all possible payoff and class distributions*.

Theorem 2: If R is the Cartesian product of the confusion matrices P and Q, then R is *more effective* than both confusion matrices P and Q.

We conclude from Theorem 2 that the maximum expected payoff that is achieved by a standard classifier could be improved, *regardless of payoff or class distribution information*, through Cartesian composite classifier architectures that incorporate additional component classifiers.[5]

III. Optimal Investment in KDD

A. Relationship Between Investment and Payoff

The basic model presented in Section II describes an environment in which the decision-maker creates a decision rule to optimize the expected payoff given a confusion matrix, a payoff function, and prior probabilities of actual classes.

In this section, we extend the basic model by defining a subspace of square and symmetric confusion matrices whose values reflect the quality of classification as a function of its cost. Larger investments in the KDD process will typically provide the decision-maker with classification of a higher quality. For example, the decision-maker would like to know how much to invest in a KDD process in order to support a credit screening application for credit cards. In order to increase the expected payoff, the credit company would like to base its decision whether to approve an applicant or not based on a data mining process with low class-conditional error rates. The quality of the data mining process, however, may become higher as the financial investments increases.

Next, we show how to incorporate the investment cost within the basic decision-theoretic framework introduced in Section II. Let C denotes the investment cost of the KDD process, and assume that the data mining measure of performance is defined in terms of a confusion matrix $P(C)$ of size $n \times n$, where

[4] The Cartesian composite classifier can be defined over several component classifiers, e.g., $R = (P \times Q) \times S$, etc. Also, recall that our model allows for classifiers, where the set of predicted classes is **not necessarily identical** to the set of actual classes. This is especially relevant (as in the above case) when considering multiple classifier combination.

[5] The final decision whether to deploy the Cartesian composite classifier or not is an investment decision, which depends on the *deployment costs* of the additional component classifiers. This crucial issue is discussed in Section III.

$n = n_s = n_y$. We further assume that the confusion matrix $P(C)$ has the following form:[6]

$$P(C) = \begin{pmatrix} f(C) & \frac{1-f(C)}{n-1} & \cdots & \frac{1-f(C)}{n-1} \\ \frac{1-f(C)}{n-1} & f(C) & \cdots & \frac{1-f(C)}{n-1} \\ \vdots & \vdots & \ddots & \\ \frac{1-f(C)}{n-1} & \cdots & & f(C) \end{pmatrix}$$

where $f(C)$ satisfies the following properties.

- For every $C \geq 0$, $1/n < f(C) \leq 1$.
- $f(C)$ is strictly increasing in C.

The reason that the diagonal elements, as well as the off-diagonal elements of $P(C)$ are equal, reflects the uniform prior probabilities that the decision-maker assigns to the class-conditional errors. Note also that the off-diagonal elements decreases when the investment cost increases, thus reflecting the fact that a larger investment will provide the decision-maker with classification of a higher quality. In practice, the function $f(C)$ may be determined by applying a procedure of fitting a parametric function to historical data relating the error rates $1 - f(C)$ of similar data mining processes to their investment costs C.

Intuitively, the decision-maker is willing to invest more for a KDD process if this would ensure a better process in terms of the relationship "*more effective*" presented in Definition 1. With a better KDD process, the decision-maker increases the expected payoff, or at least does not worsen it. The following results examine the relationship between the cost C and the effectiveness of $P(C)$. It will be shown that as the decision-maker invests more in the KDD process, a "*more effective*" confusion matrix is obtained (yielding no less expected payoff *regardless of payoff or prior probabilities information*).

Theorem 3: Let $Q(C_Q)$ and $R(C_R)$ be two confusion matrices that are used to classify examples coming from the *same* set of actual classes. $Q(C_Q)$ is more effective than $R(C_R)$ if $C_Q > C_R$.

Theorem 4: Let $Q(C_Q)$ $R(C_R)$ and P be three confusion matrices that are used to classify examples coming from the *same* set of actual classes. Let $Q^{\times}(C_Q) = Q(C_Q) \times P$ and $R^{\times}(C_R) = R(C_R) \times P$. $Q^{\times}(C_Q)$ is more effective than $R^{\times}(C_R)$ if $C_Q > C_R$.

B. Improving a KDD Process With Cartesian Composite Classification

Often, the decision-maker already employs a KDD process associated with a confusion matrix P. By Theorem 2, the decision-maker can improve the quality of the overall process by investing in an independent KDD process associated with a confusion matrix P^*. Moreover, Theorems 3 and 4 show that investing more in the second KDD process renders the overall process more effective. Formally, the Cartesian process is given by $P^{\times}(C) = P^*(C) \times P$. The expected payoff obtained from the improved process is given by

$$EU^{\times}(C, D^{\times}) = \text{trace}(P^{\times}(C)D^{\times}U\Pi) \quad (3)$$

where C is the investment cost in the second KDD process, and $D^{\times}$ is the decision rule for the Cartesian process. The *expected net payoff* $EP^{\times}$ of the decision-maker is, thus, obtained as follows:

$$EP^{\times}(C, D^{\times}) = \text{trace}(P^{\times}(C)D^{\times}U\Pi) - \text{trace}(PDU\Pi) - C. \quad (4)$$

The decision-maker's goal is to maximize the *expected net payoff* $EP^{\times}$ by stating an optimal decision rule $D^{\times}$, as well as an optimal investment cost C in the second KDD process.

Example 4: One of the most successful applications of data mining is performed in "database marketing" [19]–[21]. Database marketing is a method that enables marketers to develop customized marketing strategies based on extracted patterns derived from customer databases [19]. For example, by employing database marketing, local retailers can reach customers with the "best fit" offer and products at the right time and geographical area. As another example, telephone companies have identified and segmented high-valued customers (called "power users"). Data mining is, then, used to determine which terms and products to offer to people in this high-valued segment.

In this section, we present an example in which the marketers of a chain store wish to develop a marketing strategy that utilizes the knowledge resulting from data mining. To increase the number of sales and the amount of customer satisfaction, the marketers want to make sales promotion offers by direct mails to selected customers. In order to increase the expected payoff, the marketers determine which customer classes in a list to mail to based on patterns extracted from customer information originating from sales transactions.

The customers are of three *actual classes*: $S = \{$"loyal customer"; "regular customer"; "inconsistent customer"$\}$. The vector $\pi = (\pi_1(t), \pi_2(t), \pi_3(t))$ of prior probabilities of the actual classes under study *varies quarterly* over time due to seasonal trends and events that might affect the chain store (e.g., competition). The values of the prior probabilities (assigned by the marketers) in each quarter are presented in Table I.

The decision-maker employs a KDD process, which includes a classification-based data mining methodology based on decision trees (e.g., [12]). In particular, the customers give rise to certain attributes, which together form the attribute vector X. Following [19] and [20], the attribute vector contains the following information:

1) time period since the last purchase;
2) number of purchases made in a certain time period;
3) amount of money spent during a certain period of time.

Based on the observed value $X = x$, the task of the data mining system is to classify the customers to one of three predicted classes in $Y = \{$"loyal customer"; "regular customer"; "inconsistent customer"$\}$. The decision-maker observes predicted classes as determined by the data mining system and chooses actions accordingly. Let $A = \{$"make sales promotion offers

[6] This restriction is needed in the analysis below. However, in applications, the general case of a *nonsymmetric* confusion matrix may be considered as well (see [26]). The *nonsymmetric* case might arise when, for instance, there is prior knowledge related to misclassification errors. In the nonsymmetric case, the confusion matrices can be extracted from the output of any commercial data-mining algorithm (e.g., [18]).

TABLE I
VARIABLE PRIOR PROBABILITIES OF CUSTOMER ACTUAL CLASSES OVER TIME (QUARTERLY)

Year	1				2			
Quarter	1	2	3	4	5	6	7	8
π_1	0.7	0.6	0.55	0.62	0.8	0.75	0.65	0.85
π_2	0.2	0.25	0.25	0.28	0.1	0.15	0.2	0.08
π_3	0.1	0.15	0.2	0.1	0.1	0.1	0.15	0.07

by direct mail"; "do not make sales promotion offers by direct mail"} be the set of actions that are taken by the decision-maker. The payoff matrix U that associates payoffs with pairs of actions and actual classes is

$$U = \begin{array}{cc} & \textbf{Actual Classes} \\ \textbf{Actions} & \begin{pmatrix} -200, & 20 & 40 \\ 0, & -100 & -500 \end{pmatrix} \end{array}.$$

In the payoff matrix, the payoff for the chain store is positive when it makes sales promotion offers to "inconsistent" or "regular" customers (i.e., $u_{12} = 20, u_{13} = 40$). The payoff is negative when the chain store does not make sales promotion offers to "inconsistent" or "regular" customers (i.e., $u_{22} = -100, u_{23} = -500$). The payoff is also negative when the chain store makes sales promotion offers to "loyal" customers (i.e., $u_{11} = -200$). All the payoff values represent the total return during one quarter.

Based on past experience, the data mining measure of performance is estimated in terms of the following *confusion matrix*:

$$P = \begin{array}{cc} & \textbf{Predicted Classes} \\ \textbf{Actual Classes} & \begin{pmatrix} 0.7, & 0.15 & 0.15 \\ 0.15, & 0.7 & 0.15 \\ 0.15, & 0.15 & 0.7 \end{pmatrix} \end{array}.$$

The chain store is willing to improve the current data mining process by employing an additional independent data mining process and combining them using the Cartesian product introduced in Section II. The second process would be a "second opinion" to that provided by the chain store's current data mining process [1].

The confusion matrix of the additional data mining system as a function of the investment cost can be evaluated in several ways. Based on historical data, the following information can be extracted: 1) the investment cost per data mining application and 2) the total prediction error, which can be obtained from any commercial data-mining algorithm. Based on this data, a curve can be fitted, where the total error can be obtained as a function of investment. This of course is only an approximation, but nevertheless, it can be useful in sound decision-making. If historical data is abundant, individual error probabilities (i.e., the elements in the confusion matrix) can also be expressed as a function of the investment cost.

Another approach to gauging the dependence between the error probabilities and the investment cost is as follows. It is known that the error probabilities depend on the size of the training set. In turn, collecting and cleansing a larger data set (especially when experiments are involved, see [29]) will increase the investment cost. This provides a mechanism by which to gauge the desired functional form.

By applying a procedure of fitting a parametric function to historical data, it is found that the error rate varies with investment costs approximately as $0.6/(C+1)^{0.2}$. Thus, the second process is associated with the following confusion matrix $P(C)$:

$$P(C) = \begin{pmatrix} 1 - \dfrac{0.6}{(C+1)^{0.2}} & \dfrac{0.3}{(C+1)^{0.2}} & \dfrac{0.3}{(C+1)^{0.2}} \\ \dfrac{0.3}{(C+1)^{0.2}} & 1 - \dfrac{0.6}{(C+1)^{0.2}} & \dfrac{0.3}{(C+1)^{0.2}} \\ \dfrac{0.3}{(C+1)^{0.2}} & \dfrac{0.3}{(C+1)^{0.2}} & 1 - \dfrac{0.6}{(C+1)^{0.2}} \end{pmatrix}.$$

The Cartesian process is given as follows:

$$P^{\times}(C) = \begin{pmatrix} 1 - \dfrac{0.6}{(C+1)^{0.2}} & \dfrac{0.3}{(C+1)^{0.2}} & \dfrac{0.3}{(C+1)^{0.2}} \\ \dfrac{0.3}{(C+1)^{0.2}} & 1 - \dfrac{0.6}{(C+1)^{0.2}} & \dfrac{0.3}{(C+1)^{0.2}} \\ \dfrac{0.3}{(C+1)^{0.2}} & \dfrac{0.3}{(C+1)^{0.2}} & 1 - \dfrac{0.6}{(C+1)^{0.2}} \end{pmatrix} \times \begin{pmatrix} 0.7 & 0.15 & 0.15 \\ 0.15 & 0.7 & 0.15 \\ 0.15 & 0.15 & 0.7 \end{pmatrix}.$$

Next, the following questions are addressed. 1) What is the amount to be invested in an independent data mining process in each quarter; 2) what would be the expected net payoff after the investment is made? Following (3), let $EU^{\times}(C, D^{\times}, t)$ denote the expected payoff from the improved Cartesian process during quarter t when a decision rule $D^{\times}$ and an investment cost C in the second KDD process are chosen. As proved in Theorem 4, for each quarter t the *maximum* value of $EU^{\times}(C, D^{\times}, t)$ (obtained by choosing an *optimal* decision rule $D^{\times}$) increases as the decision-maker increases the investment cost C. Following (4), let $EP^{\times}(C, D^{\times}, t)$ denote the expected *net* payoff from the improved Cartesian process during quarter t, when a decision rule $D^{\times}$ and an investment cost C in the second KDD process are chosen. The *maximum* expected *net* payoff $EP^{\times}(C, t)$ is obtained by selecting an optimal decision rule $D^{\times}$. This optimal decision rule is determined by solving the linear programming problem (2) presented in Section II-A.

In Fig. 1, we present for each quarter $t = 1, 2, \ldots, 8$ the *maximum* expected *net* payoff $EP^{\times}(C, t)$ versus the investment cost C. For each quarter t, the optimal amount C to be

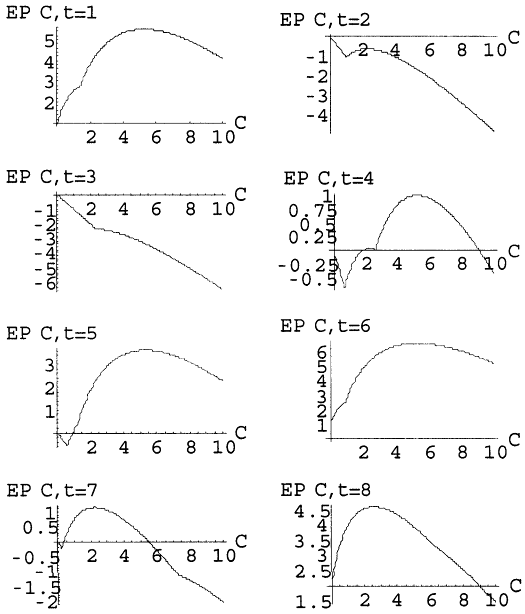

Fig. 1. For each quarter $t = 1, 2, \ldots, 8$, the maximal expected net payoff (obtained by selecting optimal decision rule $D^{\times}$) versus the investment cost C. The optimal investment cost for each quarter is obtained at a unique point.

invested in the independent KDD process is determined, then, as the investment cost that maximizes $EP^{\times}(C, t)$. For example, the optimal investment in the first quarter is $C = 5.1$, the corresponding *maximum* expected net payoff is 5.2, and the optimal decision rule is

$$D^{\times}(t=1,\, C=5.1) = \begin{array}{cc} & \textbf{Actions} \\ \textbf{Predicted Classes} & \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} \end{array}.$$

Notice that the optimal investment in the second and third quarters is $C = 0$, which means that it is not worthwhile to deploy the new data mining system in these quarters.

The sensitivity of the decision-based data mining approach to variations in the data (e.g., prior and error probabilities) can be addressed analytically as follows. We note that finding the optimal decision rule, as well as finding the optimal payoff, is equivalent to solving a related linear programming problem [see problem (2)]. Incorporating variations in the input data, it is observed that the set of constraints in the related linear programming problem remains fixed, whereas the coefficients of the objective function become a function of the input data. The sensitivity of our approach, thus, can be addressed by exploring the local and global sensitivity analysis of the underlying linear programming problem (see [28]). As an example, consider that the only source of variation is related to the payoffs. Then, the theory of linear programming says that there is a convex set in the space of payoff values where any "point" within that set will not change the optimal decision rule of the related linear programming problem. Moreover, the optimal payoff as a function of the "perturbation" can be shown to be a piecewise linear concave function. Thus, we expect that our approach is robust to variations in input data. We intend to further analyze this issue in future work.

IV. Combining Disjoint Data Mining Systems

In Section II, we have shown how several independent data mining processes that are used to classify examples coming from the *same* set of actual classes can be combined using the Cartesian operator into one data mining process. In this section, we address cases where the decision-maker applies independent data mining processes that are used to classify examples coming from *disjoint* sets of actual classes. We say that such data mining processes are *disjoint*. To illustrate, we might use a decision tree classifier used to classify examples coming from the actual "credit risk" classes *good* or *bad* and a neural network classifier used to classify examples coming from the actual "credit usage" classes *heavy* or *light*. We show how to combine two data mining processes used to classify examples coming from disjoint and independent sets of actual classes by introducing the *doubly cartesian* operator. We prove that the effectiveness of the doubly cartesian process is *not less* than any of the component data mining processes from which the doubly cartesian process is formed. We also show that improving one of the data mining processes provides more effective process for the decision-maker, regardless of the quality of the component data mining processes.

A. Doubly Cartesian Composite Classification

Next, we show how several independent data mining processes used to classify examples coming from *disjoint* sets of actual classes can be combined into one data mining process. First, we define the *doubly cartesian product* of two matrices A and B of size $n_1 \times m_1$ and $n_2 \times m_2$, respectively:

$$A \otimes B = \begin{pmatrix} a_1 \times b_1 \\ a_1 \times b_2 \\ \vdots \\ a_1 \times b_{n_2} \\ \vdots \\ a_{n_1} \times b_1 \\ a_{n_1} \times b_2 \\ \vdots \\ a_{n_1} \times b_{n_2} \end{pmatrix}$$

where a_i (respectively b_i) is the ith row in A (respectively B), and $\times$ is the *Cartesian product* defined in Section II.

Consider two independent data mining processes. Let P and Q denote the associated confusion matrices of size $n_s^P \times n_y^P$ and $n_s^Q \times n_y^Q$, respectively. The two independent data mining processes can be combined into one data mining process, called the *doubly cartesian process*, whose confusion matrix R of $n_s^P \cdot n_s^Q$ rows and $n_y^P \cdot n_y^Q$ columns is defined as follows:

$$R = P \otimes Q.$$

The doubly cartesian process is used to classify examples coming from the set of actual classes in $S^P \times S^Q$, and it classifies an example to one of the predicted classes in the set $Y^P \times Y^Q$.

Example 5: Consider two independent classifiers represented by the following confusion matrices

$$P = \begin{array}{c} \\ s_1^P \\ s_2^P \end{array} \!\! \begin{array}{cc} y_1^Q & y_2^Q \\ \multicolumn{2}{c}{\begin{pmatrix} 0.8, & 0.2 \\ 0.3, & 0.7 \end{pmatrix}} \end{array}, \qquad Q = \begin{array}{c} \\ s_1^Q \\ s_2^Q \end{array} \!\! \begin{array}{cc} y_1^Q & y_2^Q \\ \multicolumn{2}{c}{\begin{pmatrix} 0.6, & 0.4 \\ 0.2, & 0.8 \end{pmatrix}} \end{array}.$$

The doubly cartesian composite classifier is used to classify examples coming from the set of actual classes $S^{D\text{-Cartesian}} = \{s_1^P \,\&\, s_1^Q,\, s_1^P \,\&\, s_2^Q,\, s_2^P \,\&\, s_1^Q,\, s_2^P \,\&\, s_2^Q\}$. The task of the classifier is to classify an example to one of the *four* classes in $Y^{D\text{-Cartesian}} = \{y_1^P \,\&\, y_1^Q,\, y_1^P \,\&\, y_2^Q,\, y_2^P \,\&\, y_1^Q,\, y_2^P \,\&\, y_2^Q\}$ on the basis of the observed value $X = x$. The confusion matrix associated with the Doubly-Cartesian composite classifier is

$$R = P \otimes Q$$
$$= \begin{array}{c|cccc} & y_1^P \,\&\, y_1^Q & y_1^P \,\&\, y_2^Q & y_2^P \,\&\, y_1^Q & y_2^P \,\&\, y_2^Q \\ \hline s_1^P \,\&\, s_1^Q & 0.48, & 0.32 & 0.12 & 0.08 \\ s_1^P \,\&\, s_2^Q & 0.16, & 0.64 & 0.04 & 0.16 \\ s_2^P \,\&\, s_1^Q & 0.18, & 0.12 & 0.42 & 0.28 \\ s_2^P \,\&\, s_2^Q & 0.06, & 0.24 & 0.14 & 0.56 \end{array}.$$

The doubly cartesian composite classifier is implemented as follows. An example (say with actual classes s_2^P and s_1^Q) is classified by both classifiers, and their predictions are concatenated to form one of the four classes in $Y^{D\text{-Cartesian}}$. The decision-maker observes the predicted class (say $y_1^P \,\&\, y_2^Q$), and chooses the *optimal action* (say a_1) accordingly from within the set of available actions $A^{D\text{-Cartesian}} = \{a_1, a_2\}$. A payoff $u_{1,3}$ is, then, associated with the action a_1 and actual class "$s_2^P \,\&\, s_1^Q$."

Example 6: Consider a credit company that applies two independent and disjoint data mining processes. The first process is used to classify examples coming from the following set of actual classes: $S^Q = \{$"good credit risk," "bad credit risk"$\}$. The respective prior probabilities are $\pi^Q =$ (0.4 for "good credit risk," 0.6 for "bad credit risk"). The second data mining process is used to classify examples coming from the following set of actual classes: $S^R =$ ("very heavy credit usage," "heavy credit usage," "light credit usage"$\}$ with corresponding prior probabilities $\pi^R =$ (0.2 for "very heavy credit usage," 0.5 for "heavy credit usage," 0.3 for "light credit usage"). The confusion matrix of the first data mining process is

$$Q = \begin{pmatrix} 0.7, & 0.3 \\ 0.3, & 0.7 \end{pmatrix}.$$

The confusion matrix of the second data mining process is

$$R = \begin{pmatrix} 0.8, & 0.1 & 0.1 \\ 0.1, & 0.8 & 0.1 \\ 0.1, & 0.1 & 0.8 \end{pmatrix}.$$

The doubly cartesian product $Q \otimes R$ of these matrices is defined over the Cartesian product $S^Q \times S^R$ of actual classes, and Cartesian product $Y^Q \times Y^R$ of predicted classes:

$$\begin{pmatrix} 0.56 & 0.07 & 0.07 & 0.24 & 0.03 & 0.03 \\ 0.07 & 0.56 & 0.07 & 0.03 & 0.24 & 0.03 \\ 0.07 & 0.07 & 0.56 & 0.03 & 0.03 & 0.24 \\ 0.24 & 0.03 & 0.03 & 0.56 & 0.07 & 0.07 \\ 0.03 & 0.24 & 0.03 & 0.07 & 0.56 & 0.07 \\ 0.03 & 0.03 & 0.24 & 0.07 & 0.07 & 0.56 \end{pmatrix}.$$

The vector of prior probabilities of the doubly cartesian process $\pi^{Q\otimes R}$ is constructed as follows:

$\pi^{Q\otimes R} = (\pi_1^Q \cdot \pi_1^R,\, \pi_1^Q \cdot \pi_2^R,\, \pi_1^Q \cdot \pi_3^R, \pi_2^Q \cdot \pi_1^R,\, \pi_2^Q \cdot \pi_2^R,\, \pi_2^Q \cdot \pi_3^R) = (0.08, 0.2, 0.12, 0.12, 0.3, 0.18)$. The square matrix $\Pi^{Q\otimes R}$ is obtained by placing the vector of prior probabilities $\pi^{Q\otimes R}$ in the main diagonal and zeros elsewhere.

The decision-maker applies a data mining process, and chooses whether to approve an applicant or not based on the doubly cartesian data mining classification. The payoff matrix U is given by

$$U = \begin{pmatrix} 10, & -15 & 17 & -30 & 5 & 9 \\ -5, & -4 & 10 & 14 & 2 & -1 \end{pmatrix}.$$

The maximum expected payoff obtained by solving the linear programming problem (2) presented in Section II-A is

$$EU(Q \otimes R) = \text{trace}(P^{Q\otimes R} \cdot D^{\otimes} \cdot U \cdot \Pi) = 3.0004$$

where the optimal decision rule $D^{\otimes}$ is

$$D^{\otimes} = \begin{pmatrix} 1, & 0 \\ 0, & 1 \\ 1, & 0 \\ 0, & 1 \\ 0, & 1 \\ 1, & 0 \end{pmatrix}.$$

B. Effectiveness of the Doubly Cartesian Composite Classification

The following result is shown.

Theorem 5: If R is the doubly cartesian product of the confusion matrices P and Q, then R is *more effective* than both confusion matrices P and Q.

We conclude from Theorem 5 that an investment in an additional data mining process, which is combined with the current process by the doubly cartesian product, renders the overall process more effective; i.e., the expected payoff increases *regardless of payoff or actual class distribution information* Given a doubly cartesian process, the decision-maker may wish to improve one or more of the constituent processes. Theorem 6 below shows that improving the quality of any one of the constituent processes improves the effectiveness of the overall process as well.

Theorem 6: Let Q, R be two confusion matrices that are used to classify examples coming from the same set of actual classes, and P be a confusion matrix that is used to classify examples coming from a disjoint set of actual classes. Let $Q^{\otimes} = Q \otimes P$ and $R^{\otimes} = R \otimes P$. $Q^{\otimes}$ is more effective than $R^{\otimes}$ if Q is more effective than R.

V. Summary

Data mining is the process of extracting valid, previously unknown, comprehensible information from large databases and using it to make crucial organization decisions. Global, national, and even local organizations are driven by information, which is uncovered by the data mining process. Nowadays, data mining has become an essential core of KDDs and therefore, their quality must be improved as much as possible in order to guarantee successful KDD processes. Although evaluating the quality of the data mining process is one of the most pressing challenges facing KDD research today, few organizations have effective ways of managing data mining quality, which is so important to their competitiveness.

This paper considers data mining quality as a main goal to achieve, instead of a subproduct of database creation and KDD development processes. To this end, we developed a decision-theoretic approach for evaluating data mining systems, which employ classification methods, in terms of their utility in decision making. The decision-theoretic model was developed in order to provide an economic perspective on the value of “extracted information,” in terms of its payoff to the organization, and to suggest a wide range of decision problems that arise from this point of view. In the decision-based approach, the decision-maker observes predicted classes as determined by the data mining classification system and chooses actions accordingly. The decision-maker wishes to maximize the expected payoff by choosing an optimal decision rule.

The decision-theoretic framework enables us to rigorously define and analyze the concept of *actionability*—the ability of the extracted information to suggest concrete and profitable action by the decision-makers [9], [14], [16]. According to the proposed framework, data mining systems can be compared by their *effectiveness*. With a more effective data mining system, the decision-maker increases the expected payoff or at least does not worsen it.

The relation between the *quality* of a data mining system and the amount of investment that the decision maker is willing to make is formalized. The modeling relationship between investment cost and quality results in a subspace of data mining systems whose confusion matrices can be rank ordered according to their effectiveness. This also captures the intuition that the decision-maker is willing to invest more in a data mining system if this would ensure a better data mining system in terms of the “effectiveness” relation.

We proposed two ways by which independent data mining systems can be combined. The first mode of combination—by the Cartesian product—applies to independent data mining systems used to classify examples coming from the *same* set of actual classes. The second mode of combination—by the doubly cartesian product—applies to independent data mining processes used to classify examples coming from *disjoint* sets of actual classes. In both cases, we showed that the resulting data mining system is *not less* effective than any of the constituent systems. Moreover, it was shown that improving the quality of any one of the constituent processes improves the effectiveness of the overall process as well. The combination of two data mining systems can be viewed as joining an existing system with a secondary system that provides a “second opinion” about the classification of examples. With this point of view, the decision-maker’s optimization problem is to choose the optimal investment in the secondary data mining system in order to *maximize* the *expected net payoff* conveyed by the overall (combined) system.

The work presented in this paper can be extended in several ways.

1) In many data mining systems, the data is not stationary, but rather changing and evolving [7]. This changing data may make previously discovered classifications invalid. For example, in financial markets, the rapidly changing market conditions may make previously discovered patterns invalid. Moreover, the various tools and data warehouses that support the entire KDD process may change as well. Thus, it would be desirable to formulate the relationship between the data mining quality, the organization strategy, and the organization dynamic environment.
2) In certain data mining systems, several mining methods with different goals may be applied *successively*, or in a *distributed* manner in order to achieve a desired goal. For example, the organization may first apply a clustering method to segment the customer database based on their credit usage and then apply a decision tree method to classify credit risks to each cluster. The decision-theoretic framework may be extended to capture the relationship among distributed data mining systems. The extended model would make it possible to determine the optimal investment and its allocation to each data mining system in the distributed environment.
3) In this paper, we analyzed data mining methods which fall under the category of classification. We would like to extend the decision-based framework for evaluating other data mining methods (e.g., clustering see [8]) in terms of their utility in decision-making.
4) We want to develop estimation procedures for assessing the prior probabilities of actual classes and the payoff matrix associated with pairs of actions and actual classes. For example, classical statistics and machine learning methodologies can be used for analyzing the effect of marketing strategies of sending direct mails to target customers on sales increases of the organization.
5) The investment cost C was considered as a monolithic parameter. A more refined modeling of the investment cost that takes into account various factors could be a viable research direction.
6) Finally, because of the growing complexity of KDDs, we believe that the proposed approach presented in this paper can also address continuous assessment of database and data-warehouses quality throughout the overall KDD development process.

REFERENCES

[1] N. Ahituv and B. Ronen, “Orthogonal information structures—A model to evaluate the information provided by second opinion,” *Dec. Sci.*, vol. 19, pp. 255–268, July 1988.

[2] R. Brachman and T. Anand, “The process of knowledge discovery in databases: A human-centered approach,” in *Advances in Knowledge Discovery and Data Mining*, U. Fayyad, G. Piatetsky-Shapiro, S. P. Amith, and R. Uthurusamy, Eds. Cambridge, MA: MIT Press, 1996, pp. 37–58.

[3] J. S. Demski, *Information Analysis Reading*. Reading, MA: Addison-Wesley, 1972.

[4] U. Fayyad, “Data mining and knowledge discovery: Making sense out of data,” *IEEE Expert*, vol. 11, pp. 20–25, Oct. 1996.

[5] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery: An overview,” in *Advances in Knowledge Discovery and Data Mining*, U. Fayyad, G. Piatetsky-Shapiro, S. P. Amith, and R. Uthurusamy, Eds. Cambridge, MA: MIT Press, 1996, pp. 1–36.

[6] ——, “The KDD process for extracting useful knowledge from volumes of data,” *Commun. ACM*, vol. 39, no. 11, pp. 27–34, Nov. 1996.

[7] M. Goebel and L. Gruenwald, “A survey of data mining and knowledge discovery software tools,” *SIGKDD Explorations*, vol. 1, no. 1, pp. 20–33, June 1999.

[8] J. Kleinberg, C. Papadimitriou, and P. Raghavan, “A microeconomic view of data mining,” *Knowl. Disc. Data Mining*, vol. 2, no. 4, pp. 311–324, Dec. 1998.

[9] B. M. Masand and G. Piatetsky-Shapiro, “A comparison of approaches for maximizing business payoff of prediction payoff,” *in Conf. Proc. Knowledge Discovery Data Mining*, pp. 195–201, 1996.

[10] C. B. McGuire and R. Radner, *Decision and Organization*. Minneapolis, MN: Univ. of Minnesota Press, 1986.

[11] J. Marschak, “The economics of information systems,” in *Frontiers of Quantitative Economics*, M. Intrilligator, Ed. Amsterdam, The Netherlands: North-Holland, 1971, pp. 43–107.

[12] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[13] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan-Kaufmann, 1988.

[14] G. Piatetsky-Shapiro and C. J. Matheus, “The interestingness of deviations,” in *Proc. Knowledge Discovery Data Mining*, 1994, pp. 25–36.

[15] L. H. Setiono and H. Liu, “Effective data mining using neural networks,” *IEEE Trans. Know. Data Eng.*, vol. 8, pp. 957–961, Dec. 1996.

[16] A. Silberschatz and A. Tuzhilin, “What makes patterns interesting in knowledge discovery systems,” *IEEE Trans. Know. Data Eng.*, vol. 8, pp. 970–974, Dec. 1996.

[17] J. R. Quinlan, “Induction of decision trees,” *Mach. Learn.*, vol. 1, pp. 81–106, 1986.

[18] M. J. Berry and G. Linoff, *Data Mining Techniques: For Marketing, Sales, and Customer Support*. New York: Wiley, 1997.

[19] J. R. Bult and T. Wansbeek, “Optimal selection for direct mail,” *Marketing Sci.*, vol. 14, no. 4, pp. 378–381, 1995.

[20] S. H. Ha and S. C. Park, “Application of data mining tools to hotel data mart on the intranet for database marketing,” *Expert Syst. Applicat.*, vol. 15, no. 1, pp. 1–31, July 1998.

[21] R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between sets of items in a large database,” in *Proc. SIGMOD*, June 1993, pp. 207–216.

[22] B. D. Ripley, *Pattern Recognition and Neural Networks*. New York: Cambridge Univ. Press, 1996.

[23] S. S. Anand and A. G. Buchner, *Decision Support Using Data Mining*. Englewood Cliffs, NJ: Prentice-Hall, 1997, 1998.

[24] P. Turney. Cost-sensitive learning bibliography. C Online bibliography. Comput. Sci. Eng. Dept., Lehigh Univ., Bethlehem, PA. [Online]. Available: http://home.ptd.net/~olcay/cost-sensitive.html.

[25] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. New York: Wiley, 2000.

[26] D. Braha, Y. Elovici, and M. Last, “Robust decision-theoretic classification,” Tech. Rep., Ben-Gurion Univ., Beer-Sheeva, Israel, July 2002.

[27] D. Braha, *Data Mining for Design and Manufacturing: Methods and Applications*. Norwell, MA: Kluwer, 2000.

[28] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*. Belmont, CA: Athena, 1997.

[29] D. Braha and A. Shmilovici, “Data mining for improving a cleaning process in the semiconductor industry,” *IEEE Tran. Semiconduct. Manufact.*, vol. 15, pp. 91–101, Feb. 2002.

[30] Y. Elovici and D. Braha, “A decision-theoretic approach to data mining,” Tech. Rep., Ben-Gurion University, Beer-Sheeva, Israel, Jan. 2003.

Yuval Elovici received the M.Sc. degree in computer and electrical engineering from Ben-Gurion University, Beer-Sheeva, Israel, in 1991, and the Ph.D. degree in information systems from the Tel-Aviv University, Tel-Aviv, Israel, in 2000.

He is a Lecturer at the Department of Information Systems Engineering. He is the Director of several startup companies, and he consults in the areas of computer and network security. His articles have appeared in *Microprocessing and Multiprogramming, Parallel Computing, Information Retrieval* and other professional journals. His main areas of interest are computer security, Internet security, information economics, computer architecture, and parallel and distributed systems.

Dan Braha received the Ph.D. degree in industrial engineering from Tel-Aviv University, Tel-Aviv, Israel in 1996.

He is an affiliate of the New England Complex Systems Institute (NECSI), and a senior engineering faculty member at Ben-Gurion University, Beer-Sheeva, Israel. He has been a Visiting Professor at the MIT Center for Innovation in Product Development (CIPD), and a Research Associate in the Department of Manufacturing Engineering, at Boston University, Boston, MA. One of his primary areas of research is engineering design and manufacturing. His research within engineering design focuses on developing methods to help the designer move from the conceptual phase to the realization of the physical device. He has developed a mathematical theory—the Formal Design Theory (FDT). He has published extensively, including a book on the foundations of engineering design with Kluwer Academic Publishers and an edited book on data mining in design and manufacturing, also with Kluwer. He serves on the editorial board of AI EDAM and was the editor of several special journal issues. Currently, he aims to advance the understanding of Complex Engineered Systems (CES) and arrive at their formal analysis, as well as facilitate their application.

Dr. Braha has also served on executive committees and as chair in several international conferences.