A Microeconomic View of Data Mining

Jon Kleinberg * Christos Papadimitriou† Prabhakar Raghavan‡

Abstract

We present a rigorous framework, based on optimization, for evaluating data mining operations such as associations and clustering, in terms of their utility in decision-making. This framework leads quickly to some interesting computational problems related to sensitivity analysis, segmentation and the theory of games.

Keywords: Market segmentation, optimization, clustering.

*Department of Computer Science, Cornell University, Ithaca NY 14853. Email: kleinber@cs.cornell.edu. Supported in part by an Alfred P. Sloan Research Fellowship and by NSF Faculty Early Career Development Award CCR-9701399. This work was performed in part while visiting the IBM Almaden Research Center.

†Computer Science Division, Soda Hall, UC Berkeley, CA 94720. christos@cs.berkeley.edu. Research performed in part while visiting the IBM Almaden Research Center, and supported by NSF grants CCR-9626361 and IRI-9712131.

‡IBM Almaden Research Center, 650 Harry Road, San Jose CA 95120. pragh@almaden.ibm.com

1 Introduction

Data mining is about *extracting interesting patterns from raw data.* There is some agreement in the literature on what qualifies as a "pattern" (association rules and correlations [1, 2, 3, 5, 6, 12, 20, 21] as well as clustering of the data points [9], are some common classes of patterns sought), but only disjointed discussion of what "interesting" means. Most work on data mining studies how patterns are to be extracted automatically, presumably for subsequent human evaluation of the extent in which they are interesting. Automatically focusing on the "interesting" patterns has received very limited formal treatment. Patterns are often deemed "interesting" on the basis of their *confidence and support* [1], *information content* [19], and *unexpectedness* [14, 18]. The more promising concept of *actionability* —the ability of the pattern to suggest concrete and profitable action by the decision-makers [15, 17, 18], and on the sound of it very close to our concerns in this paper— has not been defined rigorously or elaborated on in the data mining literature.

We want to develop a theory of the value of extracted patterns. We believe that the question can only be addressed in a *microeconomic* framework. *A pattern in the data is interesting only to the extent in which it can be used in the decision-making process of the enterprise to increase utility.*[1] Any enterprise faces an optimization problem, which can generally be stated as

$$\max_{x \in \mathcal{D}} f(x),$$

where $\mathcal{D}$ is the domain of all possible decisions (production plans, marketing strategies, etc.), and $f(x)$ is the *utility* or *value* of decision $x \in \mathcal{D}$. Such optimization problems are the object of study in mathematical programming and microeconomics.[2]

The feasible region $\mathcal{D}$ and the objective $f(x)$ are both comparably complex components of the problem —and classical optimization theory often treats them in a unified way via Lagrange multipliers and penalty functions [7]. However, from our point of view there is a major difference between the two: We assume that the feasible region $\mathcal{D}$ is basically endogenous to the enterprise, while the objective function $f(x)$ is a function that reflects the

[1] To quote [8], *"merely finding the patterns is not enough. You must be able to respond to the patterns, to act on them, ultimately turning the data into information, the information into action, and the action into value."*

[2] There is such an optimization problem associated with virtually every enterprise; however *in real life such problems are so involved and complex, that often nobody knows exactly their detailed formulation.* The decision-makers of the enterprise base their decisions on a very rough, approximate, and heuristic understanding of the nature and behavior of the underlying optimization problem.The fact that the details of the optimization problem being solved are nebulous and unknown to the decision-makers does not make the problem less real —or its mathematical study less useful. In fact, economic theory during this century has flourished on models such as these, in which the precise nature of the functions involved is essentially unknowable; the mathematical insights derived from the abstract problem are still valuable as heuristic guides in decision-making.

enterprise's interaction with a multitude of other agents in the market (customers, suppliers, employees, competitors, the rest of the world). That is, at a first approximation the objective function can be rewritten as

$$f(x) = \sum_{i \in \mathcal{C}} f_i(x),$$

where $\mathcal{C}$ is a set of agents or other factors influencing the utility of the enterprise. We shall be calling elements of $\mathcal{C}$ "customers." We shall be deliberately vague on what they are. There are two different possible views here: On a concrete level, we can think of them as profiles of customers and other relevant agents, about whom we have gathered relevant information by a first stage of data mining; it is this first stage that our point of view seeks to influence and direct. A more abstract, but potentially equally useful, point of view is that, alternatively, we can also think of the elements of $\mathcal{C}$ as *rows of the raw table being mined* —customers, transactions, shipments, and so on.

What makes this relevant to data mining is the following crucial assumption: We assume that the contribution of customer i to the utility of the enterprise under decision x, $f_i(x)$, is in fact a complicated function of the data we have on customer i. Let y_i denote the data we have on customer i (the ith row of the table); then $f_i(x)$ is just $g(x, y_i)$, some fixed function of the decision and the data. Hence, our problem is to

$$\max_{x \in \mathcal{D}} \sum_{i \in \mathcal{C}} g(x, y_i).$$

The conventional practice in studying such problems is to replace $\sum_{i \in \mathcal{C}} g(x, y_i)$ by $g(x, \hat{y})$, where $\hat{y}$ is some *aggregate* value[3] of the customers' data (aggregate demand of a product, aggregate consumer utility function, etc.). Such aggregation is well-known to be inaccurate, resulting in suboptimal decisions, because of *non-linearities* (non-zero second partial derivatives) in the function $g(x, y_i)$. Aggregation had been tolerated in traditional microeconomics because (1) the computational requirements otherwise would be enormous, and (2) it is difficult to obtain the data y_i. The point in data mining, in our view, is that *we now have the computational power and wealth of data necessary to attack the unaggregated optimization problem, to study the intricate ways in which correlations and clusters in the data affect the enterprise's optimal decisions.*

Our goal in this paper is to study certain aspects of data mining from these perspectives — data mining in the context of economically motivated optimization problems, with a large volume of *unaggregated* data. The framework and models that we develop from these perspectives touch on a range of fundamental issues in combinatorial optimization, linear programming, and game theory; we feel that they suggest some of the first steps in a research agenda aimed at assessing quantitatively the utility of data mining operations.

[3]We use "aggregate" in its microeconomics usage — summary of a parameter over a large population — which is related but not identical to its technical meaning in databases.

Structure of the rest of the paper

In Section 2 we present three examples which illustrate our point of view, and identify and explore its various aspects: We show by a simple example how nonlinearity is an essential aspect of interestingness; we point out that the important operations of clustering and market segmentation are affected (and in fact *defined*) by microeconomic considerations; and we indicate ways in which such considerations can affect the relational semantics of the mined database. Motivated by the second of these examples, in Section 3 we introduce a novel and interesting genre of problems, called *segmentation problems*, which capture in a crisp and stylized form the clustering aspect of data mining. One can define a segmentation problem (in fact, several versions) for any conventional optimization problem; we focus on a few natural ones with obvious data-mining flavor and interest. We show that even some of the simplest possible segmentation problems are NP-complete; however, they can be solved in time linear in the number of customers. In the Section 4 we show how optimization theory (in particular, linear programming sensitivity analysis) can be employed to develop tangible criteria of "interestingness" for data mining operations.

Up to this point in the paper, we consider a single enterprise interested in mining its data to derive value. In Section 5, we turn to the problem of two competing enterprises each trying to segment a common market, adopting a set of policies to the segments they target. Building on the classical setting of game theory, we develop a notion of *segmented matrix games* to model this setting. This quickly leads to a number of novel (and largely unsolved) issues in computational game theory.

In related work [13] in the area of discrete algorithms and complexity theory [4], we have studied approximation algorithms for some of the most basic segmentation problems that arise from our framework. This leads to interesting connections with classical problems of combinatorial optimization — such as facility location and the maximization of submodular functions — and to settings in which one can concretely analyze the power of methods such as random sampling and greedy iterative-improvement algorithms. We refer the reader to [13] for further details.

2 Three examples

We pointed out above that aggregation is especially unsatisfactory and inaccurate when the cost function $g(x, y_i)$ is nonlinear in y_i. The next two anecdote-based examples illustrate certain interesting and common kinds of nonlinearities. The third example, based on the first, illustrates some of the more subtle ways in which the semantics of the underlying database can affect our objective in searching for correlations and nonlinearities.

[4] Or, as we might say, targeted at the complexity segment of the readership!

Example 1: Beer and Diapers.[5] Suppose that a retailer stocks two products in quantities x_1 and x_2; the amounts (x_1, x_2) to be stocked are the only decision variables, bounded above by capacity: $x_1 + x_2 \leq c$. The profit margins in the two products are m_1 and m_2. We have a table with 0-1 values $(y_{1,i}, y_{i,2})$ for each customer $i \in \mathcal{C}$, indicating whether the customer will buy a unit of each of the products. That is, in this toy example demand is known deterministically.

In the first scenario, customers arrive in random order, and buy whatever part of their desired basket is available. The revenue of the enterprise is in this case a function of x_1, x_2, m_1, m_2, and the aggregate demands $Y_1 = \sum_{i \in \mathcal{C}} y_{1,i}$ and $Y_2 = \sum_{i \in \mathcal{C}} \cdot y_{i,2}$. Aggregation would not distort the optimal decision, and data mining is moot.

But suppose that the customers arrive in random order, and buy their desired basket in an all-or-nothing fashion; if not all items are available, the customer buys nothing. The expected profit for the enterprise from customer i is now of the form $B_1 \cdot y_{1,i} + B_2 \cdot y_{i,2} + B_3 \cdot y_{1,i} \cdot y_{i,2}$. Because of the nonlinear term, associations between the $y_{1,i}$ and the $y_{i,2}$ columns are now important, and the optimum decision by the enterprise depends critically on them. Aggregation will no longer do the trick, and data mining is desirable, even necessary.

We propose that *associations and correlations between attributes in a table are interesting if they correspond to nonlinear terms in the objective function of the enterprise,* that is, whenever $\frac{\partial^2 g}{\partial y_i \partial y_j} \neq 0$ for the cost function g and some attributes y_i, y_j. This sheds an interesting and novel light on data mining activities, and begs the development of a quantitative mathematical theory of "interestingness", based on mathematical programming. We start on this path in Section 4.

Example 2: Market Segmentation. Telephone companies in the U.S.A. have divided their customers into two clusters: residence and business customers; they offer different terms and prices to the two. How good is this segmentation? And are there automatic ways to discover such profitable segmentations of a market?

Suppose that an enterprise has decided to subdivide its market into two segments, and apply a different marketing strategy to each segment.[6] For the purpose of the present discussion, it is not necessary to determine in detail what such a *strategy* consists of — the enterprise may offer different terms and prices to each segment, or send a different catalog to each segment. Thus, the decision space of the enterprise is now $\mathcal{D}^2$, where $\mathcal{D}$ is the set of all possible *strategies*. For each customer i and decision $x \in \mathcal{D}$, the enterprise reaps a profit of $c_i \cdot x$; i.e. for simplicity, we are assuming the profit is linear. For each pair $(x_1, x_2) \in \mathcal{D}^2$ of strategies adopted, the enterprise subdivides the set $\mathcal{C}$ of its customers into those i for which

[5]The correlation between the amount of beer and the amount of diapers bought by consumers is one of the delightful nuggets of data mining lore.

[6]This can of course generalized to k segments, or to an undetermined number of segments but with a fixed cost associated with the introduction of each segment; see Section 3.

$c_i \cdot x_1 > c_i \cdot x_2$ (to whom the enterprise will apply strategy x_1), and the rest. That is, the function $f_i(x)$ is now

$$f_i(x) = \max\{c_i \cdot x_1, c_i \cdot x_2\},$$

which is an interesting form of non-linearity. The enterprise adopts the pair of policies (x_1, x_2) which achieves

$$\max_{(x_1,x_2)\in\mathcal{D}^2} \sum_{i\in\mathcal{C}} \max\{c_i \cdot x_1, c_i \cdot x_2\}.$$

Instead of experimenting with arbitrary plausible clusterings of the data to determine if any of them are interesting and profitable, the enterprise's data miners arrive at the optimum clustering of $\mathcal{C}$ into two sets — those presented with strategy x_1 and those presented with x_2 — in a principled and systematic way, based on their understanding of the enterprise's microeconomic situation, and the data available on the customers. We further explore segmentation, and the computational complexity of the novel problems that it suggests in Section 3, and from the standpoint of approximability in [13].

Example 3: Beer and Diapers, Revisited.[7] Let us now formulate a more elaborate data-mining situation related to that of Example 1. Suppose that a retailer has a database of past transactions over a period of time and over many outlets, involving the sales of several items. Suppose further that the database is organized as a relation with these attributes: `transaction(location, dd, mm, yy, tt, item1, item2, ..., item`n`)`. Here `location` is the particular outlet where the sale occurred, `dd, mm, yy, tt` records the day and time the sale occurred, and `item`i is the amount of item i in the transaction. We wish to data mine this relation with an eye towards discovering correlations that will allow us to jointly promote items. Analyzing correlations between columns over the whole table is a central current problem in data mining (see, for example, [12]). However, in this example we focus on a more subtle issue: *Correlations in horizontal partitions* of the table (i.e., restrictions of the relation, subsets of the rows).

In a certain rigorous sense, mining correlations in subsets of the rows is ill-advised: there are so many subsets of rows that it is very likely that we can find a subset exhibiting strong correlations between any two items we choose! Obviously, we need to restrict the subsets of rows for which correlations are meaningful. We posit that *defining the right restrictions must take explicitly into account the ways in which we plan to generate revenue by exploiting the mined correlations.* For example, suppose that the actions we contemplate are joint promotions of items at particular stores. Then the only restrictions that are legitimate are unions of ones of the form `transaction[location = 'Palo Alto']`. If, in addition, it is possible to make to target promotions at particular times of the day, then

[7]This example and the research issues it suggests are the subject of on-going joint work with Rakesh Agrawal.

restrictions of the form `transaction[location = 'Palo Alto' and 12 <tt]` are legitimate. If we can target promotions by day of the week, then even more complex restrictions such as `transaction[location= 'Palo Alto' and day-of-the-week(dd,mm,yy) = 'Monday']` may be allowed. The point is that the sets of rows on which it is meaningful to mine correlations —the *targetable restrictions of the relation*— depend very explicitly on actionability considerations.

Furthermore, the actions necessary for exploiting such correlations may conflict with each other. For example, it may be impossible to jointly promote two overlapping pairs of items, or we may have a realistic upper bound on the number of joint promotions we can have in each store. Or we may have a measure of the expected revenue associated with each correlation we discover, and these estimates may in fact interact in complex ways if multiple actions on correlations are discovered. We wish to find a set of actions that generates maximum revenue. This point of view leads to interesting and novel optimization problems worthy of further study in both complexity and data mining.

3 Market Segmentation

Consider any optimization problem with linear objective function

$$\max_{x \in \mathcal{D}} c \cdot x.$$

Almost all combinatorial optimization problems, such as the minimum spanning tree problem, the traveling salesman problem, linear programming, knapsack, etc., can be formulated in this way. Suppose that we have a *very large* set $\mathcal{C}$ of *customers*, each with his/her own version c_i of the objective vector. We wish to partition $\mathcal{C}$ into k parts $\mathcal{C}_1, \ldots, \mathcal{C}_k$, so that we maximize the sum of the optima

$$\sum_{j=1}^{k} \left[\max_{x \in \mathcal{D}} \sum_{i \in C_j} c_i \cdot x \right]. \tag{3}$$

Problem (3) captures the situation in which an enterprise wishes to segment its customers into k clusters, so that it can target a different marketing strategy — e.g. a different advertising campaign, or a different price scheme — on each cluster. It seems a very hard problem, since it requires optimization over all partitions of $\mathcal{C}$, and therefore computation exponential in n.

Consider now the problem in which we wish to come up with k solutions $x_1, \ldots, x_k \in \mathcal{D}$ so as to maximize the quantity

$$\sum_{i \in \mathcal{C}} \max\{c_i \cdot x_j : j = 1, \ldots, k\}. \tag{4}$$

In contrast to problem (3), problem (4) can be solved exhaustively in time $O(nm^k)$, where m is the number of solutions; despite its exponential dependence on k, and its dependence

on m which is presumably substantial, it is *linear* in n, which is assumed here to be the truly large quantity. As we shall see in Section 3.2 the exponential dependence on k and the dependence on m seems inherent, even when the underlying optimization problem is extremely simple.

It is easy to see that problems (3) and (4) are equivalent. The intuitive reason is that they are max-max problems, and therefore the maximization operators commute. That is, in order to divide the customers in k segments, all we have to do is come up with k solutions, and then classify each customer to one of k segments, depending on which solution is maximum for this customer. The computational implications of this observation are very favorable, since $O(m^k n)$, the time naively needed for (4), is much better than $O(m2^{nk})$.

There is another variant of the general segmentation problem, arguably more realistic, in which we are seeking to choose k solutions $x_1, \ldots, x_k \in \mathcal{D}$ *for some integer* k *of our choice*, to minimize

$$\left[\sum_{i \in \mathcal{C}} \max\{c_i \cdot x_j : j = 1, \ldots, k\}\right] - \gamma \cdot k, \tag{5}$$

where γ is the cost of adding another solution and segment. Like problem (4), problem (5) can be solved exhaustively in time that is linear in n, with a larger dependence on m and k.

Problems (3) (or (4)) and (5) constitute a novel genre of problems, which we call *segmentation problems.* These problems are extremely diverse (we can define one for each classical optimization problem). We believe that they are interesting because they capture *the value of clustering* as a data mining operation. Clustering is an important problem area of algorithmic research that is also of significant interest to data mining —which it predates. One of the main motivations of clustering has been the hope that, by clustering the data in meaningfully distinct clusters, we can then proceed to make *independent decisions* for each cluster. To our knowledge, this is the first formalism of clustering that explicitly embodies this motivation.

3.1 Specific problems

There is no end to the problems we can define in this way: The MINIMUM SPANNING TREE SEGMENTATION PROBLEM, the TSP SEGMENTATION PROBLEM, the LINEAR PROGRAMMING SEGMENTATION PROBLEM, and so on —and at least three variants of each. There are a few of these problems, however, that seem especially natural and compelling, in view of the data-mining motivation (we only give the fixed k version of each):

HYPERCUBE SEGMENTATION: Given n vectors in $c_1, \ldots, c_n \in \{-1, 1\}^d$, and an integer k, find a set of k vectors $x_1, \ldots, x_k \in \{-1, 1\}^d$, to maximize the sum

$$\sum_{i=1}^{n} \max_{j=1}^{k} x_i \cdot c_j$$

.

This problem captures the situation in which we know the preferences of n customers on d components of a product, for which there is a binary choice for each component. We wish to develop k semi-customized versions of the product, so as to maximize the total number of customer-component pairs for which the customer likes the component of the variant he or she chooses.

Another interesting segmentation problem is

CATALOG SEGMENTATION: Given n vectors in $c_1, \ldots, c_n \in \{0,1\}^d$, and integers k and r, find a set of k vectors $x_1, \ldots, x_k \in \{0,1\}^d$ *each with exactly* r *ones*, to maximize the sum

$$\sum_{i=1}^{n} \max_{j=1}^{k} x_i \cdot c_j.$$

In this case, we know the interests of each customer, and we wish to mail k customized catalogs, each with r pages, to maximize total *hit* (i.e. the total number of pages of interest to customers).

3.2 Complexity

Even the most trivial optimization problems (e.g., maximizing a linear function over the d-dimensional ball, whose ordinary version can be solved by aligning the solution with the cost vector) become NP-complete in their segmentation versions. We summarize the complexity results below:

Theorem 3.1 *The segmentation problems (all three versions) corresponding to the following feasible sets $\mathcal{D}$ are NP-complete: (1) The d-dimensional unit ball, even with $k = 2$; (2) the d-dimensional unit L_1 ball; (3) the r-slice of the d-dimensional hypercube (the* CATALOG SEGMENTATION PROBLEM*), even with $k = 2$; (4) the d-dimensional hypercube, even with $k = 2$; (5) the set of all spanning trees of a graph G, even with $k = 2$.*

Sketch. Notice that the optimization problems underlying these problems are extremely easy: The one underlying (1) can be solved by aligning the solution with the cost vector, the one for (2) has only $2d$ vertices, the one for (3) can be solved by choosing the r most popular elements, and the one for (4) by simply picking the vertex that coordinate-wise agrees in sign with the cost vector. Since (2) has $2d$ vertices it can be solved in $O((2d)^k n)$ time, which is polynomial when k is bounded.

The NP-completeness reductions are surprisingly diverse: (1) is proved by a reduction from MAX CUT, (2) from HITTING SET, (3) from BIPARTITE CLIQUE, and (4) from MAXIMUM SATISFIABILITY with clauses that are equations modulo two. Finally, for SPANNING TREE

SEGMENTATION we use a reduction from HYPERCUBE SEGMENTATION (the latter problem is essentially a special case of the former, in which the graph is a path with two parallel edges between each pair of consecutive nodes).

Here we sketch only the proof of (1). Suppose that we have a graph $G = (V, E)$; direct its edges arbitrarily, and consider the node-edge incidence matrix of G (the $|V| \times |E|$ matrix with the $(i, j)^{\text{th}}$ entry equal to 1 if the j^{th} edge enters the i^{th} node, -1 if the j^{th} edge leaves the i^{th} node, and 0 otherwise). Let the $|V|$ rows of this matrix define the cost vectors $\{v_1, \ldots, v_n\}$ of the segmentation problem. Thus, we seek to divide these $|V|$ vectors into two sets, S_1 and S_2, and choose an optimal solution for each set. Let σ_i denote the sum of the vectors in the set S_i, for $i = 1, 2$. Since $\mathcal{D}$ is the unit ball, an optimal solution for S_i is simply the unit vector in the direction of σ_i, and hence the value of the solution associated with (S_1, S_2) is simply the sum of the Euclidean norms, $\|\sigma_1\| + \|\sigma_2\|$. However, it is easy to see that for any partition (S_1, S_2) of the vertices, $\|\sigma_1\| + \|\sigma_2\|$ is twice the square root of the number of edges in the cut (S_1, S_2) (because in the two sums the only entries that do not cancel out are the ones that correspond to edges in the cut); hence, solving the segmentation problem is the same as finding the maximum cut of the graph. ■

When the problem dimension is fixed, most of these problems be solved in polynomial time:

Theorem 3.2 *Segmentation problems (2–5) in the previous theorem can be solved in linear time when the number of dimensions is fixed. Problem (1) (the unit ball) can be solved in time $O(n^2 k)$ in two dimensions, and is NP-complete (for variable k) in three dimensions.*

Sketch. When the number of dimensions is a fixed constant d, the number of extreme solutions in each problem (2–5) is constant ($2d$, $\binom{d}{r}$, 2^d, and d^{d-2}, respectively). Thus the number of all possible sets of k solutions is also a bounded constant, call it c; obviously, such problems can be solved in time proportional to ckn. For (1), the 2-dimensional algorithm is based on dynamic programming, while the NP-completeness proof for $d = 3$ is by a reduction from a facility location problem. ■

4 Data Mining as Sensitivity Analysis

The field of *sensitivity analysis* in optimization seeks to develop principles and methodologies for determining how the optimum decision changes as the data change. In this section we give an extensive example suggesting that many common data mining activities can be fruitfully analyzed within this framework.

To fix ideas, we shall consider the case in which the optimization problem facing the enterprise is a *linear program* [10, 16], that is, we have an $m \times n$ matrix A (the *constraint*

matrix, $m < n$), an m-vector b (the *resource bounds*), and an n-row vector c (the *objective function coefficients*), and we seek to

$$\max_{Ax=b, x\geq 0} c \cdot x. \tag{1}$$

The columns of A —the components of x— are called *activities*, and the rows of A are called *constraints*. Extensions to non-linear inequalities and objectives are possible, with the Kuhn-Tucker conditions [7] replacing the sensitivity analysis below.

We assume, as postulated in the introduction, that the entries of A and b are fixed and given (they represent the endogenous constraints of the enterprise), while the coefficients of c depend in complex ways on a relation, which we denote by $\mathcal{C}$ (we make no distinction between the relation $\mathcal{C}$, and its set of rows, called *customers*). The ith tuple of $\mathcal{C}$ —the ith customer— is denoted y_i, and we assume that c_j is just $\sum_{i\in\mathcal{C}} f_j(y_i)$, where f_j is a function mapping the product of the domains of $\mathcal{C}$ to the reals. We noted in the introduction that the desirability of data mining depends on whether the functions f_j are "non-linear;" we shall next formalize what exactly we mean by that.

Suppose that the function $f_j(y^1, \ldots, y^r)$ satisfies $\frac{\partial^2 f_j}{\partial y^k \partial y^\ell} = 0$ for all k and ℓ. Then f_j is linear, and thus appropriate single-attribute aggregates (in particular, averages) of the table $\{y_i\}$ will accurately capture the true value of c_j. For example, if $f_j(y_1, y_2) = y_1 + 3 \cdot y_2$, then all we need in order to compute c_j is the average values of the first two columns of $\mathcal{C}$. *If* $\frac{\partial^2 f_j}{\partial y^k \partial y^\ell} \neq 0$ *for some* k *and* ℓ*, then we say that* f_j *is nonlinear.*

Assume that all attributes of the relation $\mathcal{C}$ are real numbers in the range $[0, 1]$, and that f_j depends on two attributes, call them k_j and ℓ_j (extending to more general situations is straightforward, but would make our notation and development less concrete and more cumbersome). We also assume that we have an estimate $D_j \geq 0$ on the absolute value of the derivative $\frac{\partial^2 f_j}{\partial y^{k_j} \partial y^{\ell_j}}$. $D_j > 0$ means that f_j is nonlinear. We investigate under what circumstances it is worthwhile to measure the correlations of the pair of attributes corresponding to the coefficient c_j —that is to say, to data mine the two attributes related to the jth activity. Without data mining, the coefficient c_j will be approximated by the function f_j of the aggregate values of the attributes k_j and ℓ_j.

Suppose that we have solved the linear program (1) of the enterprise, based on the aggregate estimation of the c_j's. This means that we have chosen a subset of m out the n activities, set all other activities at zero level, and selected as the optimum level of the kth chosen activity the kth component of the vector $B^{-1}b \geq 0$, where $\overline{c} = c - c_B B^{-1} A \geq 0$; here by B we denote the square nonsingular submatrix of A corresponding to the m chosen activities, and by c_B the vector c restricted to the m chosen activities. It is the fundamental result in the theory of linear programming that, if a square submatrix B is nonsingular and satisfies these two inequalities, and only then, $B^{-1}b$ is the optimum. Matrix $B^{-1}A = X$ is the *simplex tableau*, a matrix maintained by the simplex algorithm. The important question

for us is the following: *Under what changes in the c_j's will the chosen level of activities continue to be optimal?*

The theory of sensitivity analysis of linear programming [10, 16] answers this question precisely: If the c_j's are changed to new values c'_j, then the optimal solution remains the same *if and only if the condition* $c' - c'_B X \geq 0$ *is preserved,* where $X = B^{-1}A$ is the simplex tableau. That is, if each coefficient c_j is changed from its value c_j, calculated based on aggregated data, to its true value c'_j based on raw data, the optimum decision of the enterprise remains the same under the above conditions. This suggests the following quantitative measure of "interestingness" *vis à vis* data mining of the jth activity:

Definition 4.1 *For each activity j, define its* interestingness I_j *as follows:*

$$I_j = D_j \cdot (\frac{1}{\overline{c_j}} + \max_{i, X_{ij}>0} \frac{X_{ij}}{\overline{c_i}}), \tag{2}$$

where the $\overline{c_i}$'s are defined as above. Notice that both terms in I_j may be infinite; our convention is that, if $D_j = 0$, then I_j is zero.

The larger I_j is, the more likely it is that mining correlations in the attributes k_j and ℓ_j will reveal that the true optimum decision of the enterprise is different from the one arrived at by aggregation. To put it qualitatively:

An activity j is interesting if the function f_j has a highly non-linear cross-term, and either $\overline{c_j}$ is small, or the jth row of the tableau has large positive coefficients at columns with small $\overline{c_i}$'s.

As the above analysis suggests, our point of view, combined with classical linear programming sensitivity analysis, can be the basis of a quantitative theory for determining when data mining can affect decisions, ultimately *a theory for predicting the value of data mining operations.*

5 Segmentation in a Model of Competition

So far we have considered the data mining problems faced by a single enterprise trying to optimize its utility. But it is also natural to consider data mining in a setting that involves *competition* among several enterprises; to indicate some of the issues that arise, we consider the classical framework of *two-player games.* Recall that in classical game theory, such a game involves two players: I with m strategies and II with n strategies. The game is defined in terms of two $m \times n$ matrices A, B, where $A_{ij} \in \mathbf{R}$ is the revenue of Player I in the case that I chooses strategy i and II chooses strategy j; the matrix B is defined similarly in terms

of Player II. Such games have been studied and analyzed in tremendous depth in the area of game theory [4].

To add a data mining twist to the situation, suppose that two corporations I and II each have a fixed set of m and n marketing strategies, respectively, for attracting consumers. Each combination of these strategies has a different effect on each consumer, and we assume that both corporations know this effect. Thus, the set $\mathcal{C}$ of customers can be thought of as a set of $N = |\mathcal{C}|$ pairs of $m \times n$ matrices $A^1, \ldots, A^N$ and $B^1, \ldots, B^N$, where the $(i,j)^{\text{th}}$ entry of A^k is the amount won by I if I chooses strategy i and II chooses strategy j *with respect to customer k;* similarly for B^k and II.

The two corporations are going to come up with *segmented strategies*. Player I partitions the set of customers $\mathcal{C}$ into k sets, and chooses a row-strategy for each set; similarly, Player II partitions $\mathcal{C}$ into ℓ sets, and chooses a column-strategy for each set. Thus, the strategy space of the players is the set of all partitions of the N customers into k and ℓ sets, respectively. The classical theory of games tells us that there is in general a *mixed equilibrium*, in which each player selects a linear combination (which can be viewed as a probability distribution) over all possible segmentations of the market. Under these choices, each player is doing "the best he can": neither has an incentive to alter the mix of strategies he has adopted. This can be viewed as an existential result, a definition of rational behavior, rather than a concrete and efficient computational procedure for determining what each player should do.

Example 4: Catalog Wars. Suppose that the strategy space of Player I consists of all possible *catalogs* with p pages to be mailed, and similarly for Player II, where each corporation has a different set of alternatives for each page of the catalog. Each corporation knows which alternatives each given customer is going to like, and a corporation *attracts* a customer if the corporation's catalog contains more pages the customer likes than the competitor's catalog For each customer, we will say that the corporation that attracts the customer has a payoff of $+1$, while the competitor has a payoff of -1; the payoff will be zero in the case of a tie. Thus, in this hypothetical case, each individual customer is a zero-sum game between the corporations.

Each corporation is going to mail out a fixed number k of versions of the catalog, with each customer getting one version from each. The overall zero sum game has as strategy space for each player the set of all possible k-tuples of catalogs. The payoff for Player I of each pair of k-tuples, in the set-up above, is the number of customers who like more pages from some of the k catalogs of Player I than from any of the catalogs of Player II, minus the corresponding number with the roles of Players I and II interchanged.

There are many game-theoretic and computational issues suggested by such market segmentation games. Under what conditions is the existence of a pure Nash equilibrium guaranteed? What is the computational complexity of finding a mixed equilibrium?

6 Conclusions and Open Problems

We are interested in a rigorous framework for the automatic evaluation of data mining operations. To this end, we have proposed a set of principles and ideas, and developed frameworks of four distinct styles. Our framework of optimization problems with coefficients depending nonlinearly on data, and our definition of interestingness within this framework (Section 4), should be seen as an example of the kinds of theories, methodologies, and tools that can be developed; we have refrained from stating and proving actual results in that section exactly because the potential range of interesting theorems that are straightforward applications of these ideas is too broad. The segmentation problems we study in Section 3 are also meant as stylized and abstract versions of the kinds of computational tasks that emerge in the wake of our point of view. The more database-theoretic concept of *targetable restrictions* of databases introduced in Example 3 needs further exploration. Finally, we have only pointed out the interesting models and problems that arise if segmentation is seen in a context involving competition.

The technical and model-building open problems suggested by this work are too many to list exhaustively here. For example: What interesting theorems can be stated and proved within the sensitivity analysis framework of Section 4? Are there interesting and realistic segmentation problems that are tractable, or for which fast and effective heuristics can be developed? How does one model temporal issues within this framework (we can view problem faced by the enterprise as a Markov Decision Process [11]), and what insights result? More importantly, what new ideas are needed in order to apply our framework in realistic situations, in which the precise nature of the enterprise's decision problem is murky, and the data on customers incomplete, unreliable, and only very implicitly containing information on the parameters of interest (such as revenue resulting from each possible marketing strategy)?

Finally, we have focused on data mining as an activity by a revenue-maximizing enterprise examining ways to exploit information it has on its customers. There are of course many important applications of data mining — for example, in scientific and engineering contexts — to which our framework does not seem to apply directly. However, even in these applications it is possible that insight can be gained by identifying, articulating quantitatively, and taking into account the goals and objectives of the data mining activity.

Acknowledgements. We thank Rakesh Agrawal for valuable discussions.

References

[1] R. Agrawal, T. Imielinski, A. Swami. "Mining association rules between sets of items in a large database." *Proc. ACM SIGMOD Intl. Conference on Management of Data*, pp. 207–216, 1993.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. I. Verkamo. "Fast discovery of association rules." *Advances in Knowledge discovery and data mining*, pp. 307–328, AAAI/MIT Press, 1996.

[3] R. Agrawal, R. Srikant. Fast algorithms for mining association rules. *Proc. 20th Intl. Conference on Very Large Databases*, 487–499, 1994.

[4] R. Aumann, S. Hart, editors. *Handbook of Game Theory, volume I*, Elsevier, 1992.

[5] S. Brin, R. Motwani, J.D. Ullman, S. Tsur. "Dynamic itemset counting and implication rules for market basket data". *Proc. ACM SIGMOD Intl. Conference on Management of Data*, 1997.

[6] S. Brin, R. Motwani, C. Silverstein. "Beyond Market Baskets: Generalizing Association Rules to Correlations." *Proc. ACM SIGMOD Intl. Conference on Management of Data*, 1997.

[7] M. Avriel. *Nonlinear Programming: Analysis and Methods*. Prentice-Hall, 1976.

[8] M. J. Berry, G. Linoff. Data Mining Techniques. John-Wiley, 1997.

[9] M. S. Chen, J. Han, P. S. Yu. "Data mining: An overview from a database perspective." *IEEE Trans. on Knowledge and Data Eng., 8*, 6, pp. 866–884, 1996.

[10] G. B. Dantzig *Linear programming and Extensions*. Princeton Univ. Press, 1963.

[11] C. Derman. *Finite State Markov Decision Processes*. Academic Press, New York, 1970.

[12] D. Gunopoulos, R. Khardon, H. Mannila, H. Toivonen. "Data mining, hypergraph transversals, and machine learning". *Proc. ACM SIGACT-SIGMOD Symposium on Principles of Database Systems,* pp. 209–217, 1997.

[13] J. Kleinberg, C. H. Papadimitriou, P. Raghavan. "Segmentation problems," *Proc. ACM Symposium on Theory of Computing*, 1998.

[14] B. Liu and W. Hsu. "Post-analysis of learned rules." *Proc. National Conference on Artificial Intelligence*, pp. 828–834, 1996.

[15] B.M. Masand and G. Piatetsky-Shapiro. "A comparison of approaches for maximizing business payoff of prediction models". *Proc. Intl. Conference on Knowledge Discovery and Data Mining*, 1996.

[16] C. H. Papadimitriou, K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity* (second edition). Dover, 1997.

[17] G. Piatetsky-Shapiro, C. J. Matheus. "The interestingness of deviations." *Proc. Intl. Conference on Knowledge Discovery and Data Mining*, 25–36, 1994.

[18] A. Silberschatz and A. Tuzhilin. "What makes patterns interesting in knowledge discovery systems ." *IEEE Trans. on Knowledge and Data Eng., 8*, 6, 1996.

[19] P. Smyth, R. M. Goodman. "Rule induction using information theory." *Proc. Intl. Conference on Knowledge Discovery and Data Mining*, 159–176, 1991.

[20] R. Srikant and R. Agrawal. Mining generalized association rules. *Proc. Intl. Conference on Very Large Databases*, 1995.

[21] H. Toivonen. Sampling large databases for finding association rules. *Proc. Intl. Conference on Very Large Databases*, 1996.