CE:The Classifier–Estimator Framework for Data Mining

M.M. Dalkilic, E.L. Roberston, and D. Van Gucht
 Computer Science
 Indiana University
 Bloomington, Indiana 47405, USA.
 E-mail: {dalkilic, edrbtsn, vgucht}@cs.indiana.edu

Abstract

The aim of this research is to establish a coherent framework for data mining. Observing that data mining depends on two partitions, the classifier and the estimator, this paper defines the classifier/estimator (CE) framework. The classifier indicates the target of the data mining investigation. The classifier may be difficult to express from the data instance or may involve an “oracle” beyond the extant data. The estimator is typically simply expressible using the data instance. The degree to which the estimator refines the classifier partition can be used to measure how well the data instance matches the concept being investigated.

The CE framework is shown to generalize a variety of data mining and database concepts, including rough sets, functional dependency, multivalued dependency, and association rules. Furthermore, the CE framework suggests a wider range of data mining questions. The CE framework is shown to naturally express qualitative and quantitative measures of the quality of approximation. Additionally, the CE framework allows a question to be posed at a number of different conceptual scopes from local to global interests.

Keywords

Data mining, framework, partition, relational model, rough sets, association rules, functional dependencies, multivalued dependencies, error

1 INTRODUCTION

Data mining faces a predicament similar to that databases faced prior to Codd’s (Codd 1970) introduction of the relational model: different data mining problems seem to have little to do with one another, approaches are generally *ad-hoc*, there is no concise or precise means of specifying problems, and so forth. This situation has been observed elsewhere, for example (Imielinski and H.Mannila 1996, Mannila 1997). As with databases when all access was navigational, data mining semantics is often defined by implementation, in this

case search. There is no distinction between what is being sought and how the search is being carried out. While some of this admittedly is the result of data mining's diversity, there do exist enough common features to provide uniformity, if not for all of data mining, at least for a well-defined portion. The aim of this research is to establish a coherent framework for data mining; in this paper we focus on the *relational model* (Codd 1970, Abiteboul, Hull and Vianu 1995a, Ullman 1988a). Our exemplar is the success of the relational model in addressing two particular issues in databases: providing data independence and a rigorous mathematical model. We address comparable problems in data mining with the hope of achieving like-minded results.

Data mining is the elicitation of useful information from large amounts of data (Fayyad and G.Piatetsky-Shapiro 1996a). The kind of information generally falls into one of four broad categories: *facts*, *models*, *deviations*, and *trends* (Simoudis 1996). Facts are directly supported by the data and include information such as "if symptom(alpha-fetoprotein), then cancer(liver)." Modelling characterizes the data and enables the user to seek information entailed by the data, such as predicting the class to which a new data item belongs. Deviations look for data that does not belong. Trends look at data in the context of a temporal dimension, time being promoted to a basic type. Our framework supports the first three categories. An investigation is made with respect to some *concept** that varies along two dimensions. First, there is a degree of generalization/specialization, in that the mining may look for general aspects of all the data or specific details of a select subset *. Second, there is a degree of approximate validity, in that the mined information lies somewhere between true and false

To be somewhat more concrete, suppose our concept is $X \Diamond Y$, where $\Diamond$ indicates some relationship between data components X and Y . Irrespective of *how* we interpret $X, Y, \Diamond$, we can envision a degree of detail/generalization. For example, at the most general we are interested in all X, Y that might hold for $\Diamond$. In this case, X, Y are *free* and are found in some X -space and Y -space, respectively. But we are also interested in less *global* information, *e.g.*, all Y s that hold for a distinguished X . In this case, Y is *free* and X is *fixed*. The second dimension is degree of quality of approximation. Since "true" information is hard to come by, we settle for its approximation, the usefulness of which is decided by both the goodness of the approximation and our need. While specification of a good approximation depends upon the application, there is only one specification of absolutely true information, what we call *perfect* information, against which all approximations will ultimately be judged.

The next section gives a glimpse of our framework via a standard database

*A notion that, while too broad to define formally, is understood to establish a context of what one is interested in.

*Variations of this are commonly referred to as "roll-up" and "drill-down" in the *data cube* approach (Harinarayan, J.D.Ullman and Rajaraman 1996).

concept, functional dependency. Section 3 examines a popular data mining framework, rough sets, and begins to formally develop our framework; section 4 completes this formalization. Section 5 shows how some typical database and data mining concepts are characterized in terms of our framework. Section 6 develops a suite of quantitative and qualitative metrics. The remaining section gives a summary and points toward future work.

2 INTRODUCTION TO THE FRAMEWORK

We begin by choosing a concept for which to mine; we can choose any one of a number of concepts, but decide upon functional dependency (FD)(Codd 1972, Vardi 1987, Abiteboul et al. 1995a, Ullman 1988a, Codd 1972). FDs are well-known, have a formal definition, support a well-studied theory, and have proved invaluable in their practical application in many different areas. We first present an informal description of FDs. We next explain how our framework *perfectly* characterizes this concept. We then discuss how the CE framework can *vary* the two dimensions (generalization/specialization and truth approximation) of FDs.

We now present a formal definition of functional dependency in the context of a specific example; this formalism is, however, general in that replacing the specific attribute names with more abstract “ X ”, “ Y ”, or “ Z ” gives a suitably general definition of FD. The example continues the medical theme: the relation ADMIT has four attributes: `pnt`, `phy`, `date`, and `hos`. The meaning of the tuple $t \in \text{ADMIT}$ is that patient $t.\text{pnt}$ was admitted to hospital $t.\text{hos}$ on $t.\text{date}$ with physician $t.\text{phy}$ attending. A medical care manager might interested in whether every physician practices (as attending physician) at only one hospital. If this were true, it would mean that the FD $\text{phy} \rightarrow \text{hos}$ holds. Using notation such as $\text{ADMIT}.\text{phy}$ to indicate the set of all physicians in the relation ADMIT, the above FD formalizes as

$$(\forall p \in \text{ADMIT}.\text{phy})(\exists h \in \text{ADMIT}.\text{hos})(\forall t \in \text{ADMIT})[t.\text{phy} = p \rightarrow t.\text{hos} = h]. \quad (1)$$

We now describe an informal (the formalism follows later) characterization of $\text{phy} \rightarrow \text{hos}$ in the CE framework. Given an instance ADMIT for which we wish to test $\text{phy} \rightarrow \text{hos}$, we first define two classes of parameterized sets: $E_p = \{t \in \text{ADMIT} | t.\text{phy} = p\}$ and $C_h = \{t \in \text{ADMIT} | t.\text{hos} = h\}$. It is obvious that distinct p values give distinct and disjoint E_p and that each $t \in \text{ADMIT}$ belongs to one E_p —for the p such that $t.\text{phy} = p$. Thus, $\{E_p\}_{p \in \text{ADMIT}.\text{phys}}$ is a partition of ADMIT. Similarly, $\{C_h\}_{h \in \text{ADMIT}.\text{hos}}$ is a partition of ADMIT. The stage is now set, since the CE framework requires two partitions, the *classifier* and the *estimator*—in this case, $\{C_h\}_{h \in \text{ADMIT}.\text{hos}}$ and $\{E_p\}_{p \in \text{ADMIT}.\text{phys}}$, respectively. Then $\text{phy} \rightarrow \text{hos}$ holds iff every E_p is a subset of one specific C_h . Writing this symbolically, including an explicit formulation of subset, gives

Figure 1 One possible interpretation of “approximate” functional dependencies illustrated in the CE framework. Each pie piece represents a single class of the classifier, a partition on the attribute *hospital*, and corresponds to one of three possible hospitals: Mercy, Memorial, and University. Each filled ellipse represents a single class of the estimator, a partition on the attribute *phy*, and corresponds to one of three physicians, Smith, Brown, and Jones. One class of the estimator, the dashed ellipse containing Jones, lies across two classes of the classifier, a *straddler*. Suppose $|E_{Smith} \cup E_{Brown}|/|\mathbf{r}| \geq 90\%$. Then $phy \rightsquigarrow hos$.

exactly the same form as our initial sentence, *viz.*, $(\forall E_p)(\exists C_h)[E_p \subseteq C_h]$ or equivalently, $(\forall E_p)(\exists C_y)(\forall t \in ADMIT)[t \in E_p \rightarrow t \in C_h]$. This alternative formulation provides nothing to standard functional dependency theory (beyond demonstrating the adequacy of the CE framework to capture this important concept), but the classifier/estimator pair is the central feature of data mining to which we now return.

Another dimension in which an FD can be investigated is to see how well it fits the data (this is related to (J.Kivinen and Mannila 1995)). Since $phy \rightarrow hos$ is not the consequence of any business rule, there are a variety of questions the medical care manager might pose: does it, in fact, always hold true (the perfect case), does it hold for a particular physician and hospital, does it hold with a few exceptions for each physician, does it hold for most physicians but not for a small number of exceptions? The latter two cases are different varieties of approximations. Approximations are awkward to express in an *FOL* formulation of functional dependencies, but the many variations are all natural using the CE framework. The common aspect of any approximation of $X \rightarrow Y$ is that deviations are always associated with those estimator sets that split over more than one classifier, that is, the E_x that “straddle the boundary” of some C_y . Such an E_x we call a *straddler*.

One approximation of $X \rightarrow Y$ (written $X \rightsquigarrow Y$) might take as valid $X \rightsquigarrow Y$ if 90% of the tuples of $\mathbf{r}$ belong to an E_x that does not straddle any C_y —a global evaluation (see Figure 1). Another approximation criterion is local,

finding all x such that 90% of E_x belongs to the corresponding C_y . If this percentage parameter is less than 50%, this could yield an approximation to a functional dependency, *e.g.*, for *phy* and *hos*, which was no longer functional, in the sense that *Jones* $\rightsquigarrow$ *Memorial* and *Jones* $\rightsquigarrow$ *University*.

3 FROM ROUGH SETS TO CLASSIFIER/ESTIMATOR

Rough sets have become a popular framework for data mining investigations (Pawlak, Grzymala-Busse, Slowinski and Ziarko 1995, Slowinski and Stefanowski 1996, Lin 1997) but they must be generalized to capture many important database concepts that also pertain to data mining. In this section, we introduce rough sets, discuss why this framework fails to capture certain other concepts, introduce a broader framework, and show how this framework generalizes rough sets.

Rough sets were introduced by Pawlak (Pawlak 1982, Pawlak 1991) as a mathematical tool to reason about vagueness and uncertainty. Pawlak credits Frege's *boundary-line* view—that a property is vague if there exists objects for which neither the property nor its complement completely hold—as his motivation. He finds this boundary by differencing two sets: the *upper-bound* that contains objects for which the property *possibly* holds and the *lower-bound* for which the property *certainly* holds. The possibility and certainty are established by operating over partitions, requiring that the property must hold for at least one member of the entire class (possibility) or for the entire class (certainty). A rough set application attempts to approximate a *property* P , where a property is merely a subset of a finite *universe* $\mathcal{U}$. This approximation is achieved using an equivalence relation $\mathcal{E}$ over $\mathcal{U}$, establishing two bounds:

$$\text{lower-bound} \quad \wedge P_{\mathcal{E}} = \bigcup \{E | E \subseteq P, \text{ for } E \in \mathcal{E}\}. \quad (2)$$

$$\text{upper-bound} \quad \vee P_{\mathcal{E}} = \bigcup \{E | E \cap P \neq \emptyset, \text{ for } E \in \mathcal{E}\}. \quad (3)$$

Three ideas, crucial to data mining, were implicit (but not explicit) in the rough sets framework. They are:

1. Analyze a property P by investigating its relationship(s) with certain partitions of $\mathcal{U}$.
2. Evaluate the interaction of the property P and the partition $\mathcal{E}$ at the level of individual classes of $\mathcal{E}$.
3. Measure the “goodness” of approximation of P by $\mathcal{E}$ in terms of $\vee P_{\mathcal{E}} - \wedge P_{\mathcal{E}}$, that subset of $\mathcal{U}$ for which membership in an $\mathcal{E}$ -class does not determine membership or non-membership in P .

As was discussed above, functional dependency is a concept from database design theory which suggests a variety of data mining investigations. Unfortunately, FDs are completely beyond the reach of the traditional rough set

characterization because a property only provides a binary partition of $\mathcal{U}$ (namely, the partition $\{P, \bar{P}\}$).

With this in mind, we define the Classifier/Estimator (CE) framework in terms of a *classifier* $\mathcal{C}$ and an *estimator* $\mathcal{E}$. Throughout the remainder of this paper, $\mathcal{U}$ is the *universe*, a finite set; complements are taken with respect to $\mathcal{U}$ unless explicitly specified. $\mathcal{C}$ and $\mathcal{E}$ are partitions of $\mathcal{U}$, called the *classifier* and *estimator* respectively. A partition of course induces and is induced by an equivalence relation on $\mathcal{U}$; for partition $\mathcal{B}$, we write xBy to indicate that x and y belong to the same $\mathcal{B}$ -class. The use of $\mathcal{E}$ is entirely consistent with the $\mathcal{E}$ of rough sets; $\mathcal{C}$ is a generalization of the partition $\{P, \bar{P}\}$. We use $\mathcal{P}$ to denote the classifier $\{P, \bar{P}\}$.

Rather than using the concepts of upper-bound and lower-bound, which fit a single property P but not an arbitrary partition $\mathcal{C}$, the CE framework generalizes the set difference of upper-bound and lower-bound. The critical factor here is that this difference is a union of $\mathcal{E}$ -classes that *straddle* the boundary of some $\mathcal{C}$ -classes. We formalize these notions:

Definition 1 For $E \in \mathcal{E}$ and $C \in \mathcal{C}$, E *straddles* C , written $E \bowtie C$, iff $E \cap C \neq \emptyset$ and $E \cap \bar{C} \neq \emptyset$. Such an E , irrespective of the C , is called a *straddler*. ■

Our boundary then, is made up of elements from straddlers. This boundary is called the *indeterminate* set (for reasons we will give later) and is defined as follows:

Definition 2 The *indeterminate set*, written $\mathcal{I}_{\mathcal{C}}^{\mathcal{E}}$, is $\mathcal{I}_{\mathcal{C}}^{\mathcal{E}} = \bigcup \{E | E \text{ is a straddler}\}$. ■

Proposition 1 $P - \mathcal{I}_{\mathcal{P}}^{\mathcal{E}} = \bigwedge P_{\mathcal{E}}$.

$$\begin{aligned} \text{Proof } P - \mathcal{I}_{\mathcal{P}}^{\mathcal{E}} &= \bigcup (E \cap P) - \mathcal{I}_{\mathcal{P}}^{\mathcal{E}} \\ &= \bigcup \{E | E \subseteq P\} \cup \bigcup \{E | E \bowtie P\} - \bigcup \{E | E \bowtie P\} \\ &= \bigcup \{E | E \subseteq P\} \end{aligned}$$

■

Proposition 2 $P \cup \mathcal{I}_{\mathcal{P}}^{\mathcal{E}} = \bigvee P_{\mathcal{E}}$.

$$\begin{aligned} \text{Proof } P \cup \mathcal{I}_{\mathcal{P}}^{\mathcal{E}} &= \bigcup \{E \cap P | E \cap P \neq \emptyset\} \cup \\ &\quad \bigcup \{E | E \cap P \neq \emptyset \wedge E - P \neq \emptyset\} \\ &= \bigcup \{E | E \cap P \neq \emptyset \wedge E - P = \emptyset\} \cup \\ &\quad \bigcup \{E | E \cap P \neq \emptyset \wedge E - P \neq \emptyset\} \\ &= \bigcup \{E | E \cap P \neq \emptyset\} \end{aligned}$$

■

Example 1 (Rough Sets) Let $\mathcal{U} = \{t_1, t_2, t_3, t_4, t_5\}$ with an estimator $\mathcal{E} = \{\{t_1, t_2, t_4\}, \{t_3\}, \{t_5\}\}$ and binary property $\mathcal{P} = \{P, \bar{P}\}$, where $P = \{t_1, t_2, t_5\}$.

Then $\mathcal{I}_{\mathcal{P}}^{\mathcal{E}} = \{t_1, t_2, t_4\}$ and the lower and upper approximations of P are $\wedge P_{\mathcal{E}} = P - \mathcal{I}_{\mathcal{P}}^{\mathcal{E}} = \{t_5\}$ and $\vee P_{\mathcal{E}} = P \cup \mathcal{I}_{\mathcal{P}}^{\mathcal{E}} = \{t_1, t_2, t_4, t_5\}$.

$\mathcal{U}$ is partitioned into halves, the shaded portion is P and the unshaded is $\bar{P}$. The classes of $\mathcal{E}$ are demarcated by the ellipses, the dashed ellipse is a straddler.

■

Observe that, for any binary classification, there is a duality between the lower and upper bounds: $\vee P_{\mathcal{E}} = \mathcal{U} - \wedge \bar{P}_{\mathcal{E}}$. This duality is an artifact of the size of $\mathcal{P}$, since when $|\mathcal{P}| = 2$, there is only one way in which E can be a straddler. When $|\mathcal{P}| \geq 3$, however, the duality breaks down. The intuition is that lower-bounds remain stable because they are *completely* contained and therefore, any n -ary classification cannot affect this. The upper-bound of one class, however, could straddle up to $n - 1$ classes. It is for this reason we choose “indeterminate” for the boundary, since this set contains elements that are indeterminate *wrt* their membership in classes of the classifier, given the estimator.

4 INDETERMINATE SETS

Now that we have shown how the CE framework handles rough sets, we take another look at the indeterminate set.

Suppose for some $\mathcal{E}, \mathcal{C}$, that $\mathcal{I}_{\mathcal{C}}^{\mathcal{E}} = \emptyset$. We know this means there are no straddlers, but there is another meaning associated with this condition, *partition refinement*. Given two partitions $\mathcal{A}, \mathcal{B}$ over $\mathcal{U}$, $\mathcal{B}$ is a *refinement* of $\mathcal{A}$, written $\mathcal{A} \preceq \mathcal{B}$, read “ $\mathcal{A}$ is less refined than $\mathcal{B}$,” iff, for every class $B \in \mathcal{B}$, there exists a class $A \in \mathcal{A}$ such that $B \subseteq A$. We now have the following proposition:

Proposition 3 $\mathcal{I}_{\mathcal{C}}^{\mathcal{E}} = \emptyset$ iff $\mathcal{C} \preceq \mathcal{E}$.

Proof For any $E \in \mathcal{E}$, E intersects at least one $C \in \mathcal{C}$. But since $\mathcal{I}_{\mathcal{C}}^{\mathcal{E}} = \emptyset$, E is not a straddler and hence, $E \cap \bar{C} = \emptyset$ and therefore, $E \subseteq C$. Now suppose $\mathcal{C} \preceq \mathcal{E}$. Then there are no straddlers E , and therefore, $\mathcal{I}_{\mathcal{C}}^{\mathcal{E}} = \emptyset$. ■

Consider now when $\mathcal{I}_{\mathcal{C}}^{\mathcal{E}} \neq \emptyset$. Although this means that $\mathcal{C} \preceq \mathcal{E}$ does not hold, we can associate some measure of “mismatch” with the set $\mathcal{I}_{\mathcal{C}}^{\mathcal{E}}$. This in turn provides us a means of approximating how much of a mismatch occurred. We

will discuss this more at length in Section 5, Metrics, but it suffices here to say that the indeterminate set can provide both qualitative and quantitative measures in a more general way than rough sets.

5 APPLICATIONS OF THE FRAMEWORK

The goal of this section is to validate the CE framework by showing that it captures a wide variety of database and data mining themes.

In each case we will exhibit a classifier and an estimator that *perfectly characterize* the concept at hand. But just as importantly, in each case it is also possible to transform the perfect version into a range of data mining investigations varying along the breadth/specificity and approximation dimensions (as in Section 2 above). Each of those variations is implicit in some special treatment of the interaction of the $\mathcal{C}$ and $\mathcal{E}$ classes, but we do not give any variation explicitly at this point. A subsequent section, “Metrics”, discusses a toolkit for evaluating approximations.

To facilitate our discussion, let $R[A]$ denote a relational schema where $A = \{A_1, \dots, A_n\}$, and $\mathbf{r}$ an instance over $R[A]$. For $X \subseteq A$, we write $[X]$ to mean a partition of $\mathbf{r}$ into sets of tuples that agree on X . Formally $s[X]t \Leftrightarrow s.X = t.X$, for $s, t \in \mathbf{r}$.

5.1 Functional Dependencies

Dependency theory has long been used to facilitate relational database design, but dependencies also arise quite naturally and often by themselves in databases. Much study has been devoted to this area, and there are numerous important and useful results. There is, for example, a small set of inference rules (usually called *Armstrong's Axioms* (Armstrong 1974, Abiteboul, Hull and Vianu 1995b)), both sound and complete, for FDs. With these axioms, we can reason with and about FDs. Clearly, if we had a means of establishing whether some number of FDs held (perfectly or approximately), we could use these axioms to draw further, logically correct, conclusions about our data. Here we show how the CE framework establishes when an FD exists, using an equivalent formulation: $X \rightarrow Y$ iff $(\forall s \in \mathbf{r})(\forall t \in \mathbf{r})[s.X = t.X \Rightarrow s.Y = t.Y]$.

Proposition 4 $X \rightarrow Y$ iff $\mathcal{I}_{[Y]}^{[X]} = \emptyset$.

Proof Suppose $X \rightarrow Y$. We show that $E \in [X]$ does not straddle any $C \in [Y]$. Since, for every $s, t \in \mathbf{r}$, $s.X = t.X \rightarrow s.Y = t.Y$, it follows that $E \subseteq C$ for some C and $E \cap C' = \emptyset$ for any other C' , so E is not a straddler. Now, suppose $\mathcal{I}_{[Y]}^{[X]} = \emptyset$. This means for every $E \in [X]$, there exists some $C \in [Y]$ such that $E \subseteq C$ and $E \cap C' = \emptyset$, for all other $C' \in [Y]$. In other words, $s, t \in \mathbf{r}$, $s.X = t.X \rightarrow s.Y = t.Y$. Hence, $X \rightarrow Y$. ■

5.2 Multivalued Dependencies

Although data mining has largely ignored multivalued dependencies (MVDs) (Delobel 1978, Fagin 1977, Ullman 1988b), they too are an interesting kind of dependency for a number of reasons: MVDs occur quite frequently, are the “flipside” of FDs (two sets of attributes are wholly independent rather than functionally dependent), and like FDs, there exists a sound and complete set of axioms for reasoning about MVDs (Beeri, R.Fagin and J.H.Howard 1977, Ullman 1988b). Essentially an MVD requires that a set of Y values be associated with a particular X value. In fact, when the relational formalism is extended to allow sets as well as atomic values, an MVD is simply an FD with set-valued attribute on the right-hand side. Before giving the CE characterization, we formally define an MVD.

In the following, say $A = X \cup Y \cup Z$, where X, Y, Z are pairwise disjoint. We write $t \in \mathbf{r}$ as $\langle x, y, z \rangle$, where $t.X = x$, $t.Y = y$, and $t.Z = z$. Also, for tuple s , $N_y(s) = \{y | \langle s.X, y, s.Z \rangle \in \mathbf{r}\}$.

Definition 3 X *multidetermines* Y in the context Z (mention of Z is often omitted since it is implied by $Z = A - (X \cup Y)$), written $X \twoheadrightarrow Y|Z$ if whenever $s, t \in \mathbf{r}$ and $s.X = t.X$, then $N_y(s) = N_y(t)$. ■

With this formulation and $\mathcal{C}, \mathcal{E}$ pairs is given by $\mathcal{E} = [X]$ and $\mathcal{C}$ such that $s \mathcal{C} t$ iff $N_y(s) = N_y(t)$, the desired result is immediate.

Proposition 5 $X \twoheadrightarrow Y|Z$ iff $\mathcal{C} \preceq \mathcal{E}$. ■

5.3 Association Rules

Association rules (ARs) have been gaining popularity in both data mining and databases, discussed in (Agrawal, Imielinski and A.Swami 1993, Agrawal and R.Srikant 1994, Mannila, H.Toivonen and A.Inkeri Verkamo 1994, Mannila 1997). As a concept, ARs begin with an instance $\mathbf{r}$ over $R[A]$, where each attribute $A_i \in A$ has a boolean domain $\{+, -\}$. For $W \subseteq A$, we write W_+ to mean the set of tuples $\{t | t \in \mathbf{r} \wedge t.A_i = +, \text{ for each } A_i \in W\}$. Let $A = X \cup Y \cup Z$ and without loss of generality let $X = \{A_1, \dots, A_k\}$ and $Y = \{A_{k+1}, \dots, A_m\}$. We use “ $\Rightarrow$ ” to indicate an AR; in particular the expression $A_1 \wedge \dots \wedge A_k \Rightarrow A_{k+1} \wedge \dots \wedge A_m$, signifies that for $t \in \mathbf{r}$, $t \in X_+$ implies there is a *tendency* that $t \in Y_+$. This tendency is indicated by a *confidence* c and *support* s . The confidence is meant to denote strength and is the ratio $|XY_+|/|X_+|$. The support provides the overall frequency of the rule and is the ratio $|XY_+|/|\mathbf{r}|$. Rules that have high confidence and strong support are said to be *strong*. The task is to discover strong association rules. We will show later how an association rule can be handled by a local application of

FDs, but first we directly characterize AR in the CE framework. Because the issue of partition refinement is “value-blind”*, we must make certain that $\mathbf{r}$ contains a tuple t^+ with a + in every attribute—this may be accomplished *ad hoc* by adding a tag attribute, with a value “-”, to all tuples of the original $\mathbf{r}$ and then adding tuple t^+ with value “+” for all attributes, including the tag.

With this slightly constrained $\mathbf{r}$, we move to the CE definitions. The classifier is $\mathcal{C} = \{XY_+, \overline{XY}_+\}$ and the estimator is $\mathcal{E} = \{X_+, \overline{X}_+\}$.

Proposition 6 $X \Rightarrow Y, c = 1$ iff $\mathcal{C} \preceq \mathcal{E}$.

Proof Suppose $X \Rightarrow Y, c = 1$. By the definition of c , $|XY_+| = |X_+|$. Since $XY_+ \subseteq X_+$, $XY_+ = X_+$, and $\overline{XY}_+ = \overline{X}_+$. Now, suppose $\mathcal{C} \preceq \mathcal{E}$. Since $t_+ \in XY_+ \cap X_+$, this can only hold if $XY_+ = X_+$. This gives $c = 1$ as required. ■

An AR is strongly related to an FD, i.e. $\overbrace{+, \dots, +}^X \rightarrow \overbrace{+, \dots, +}^Y$ in the example above. Using localization of FDs allows exploration of significant associations where some attributes are negative as well as some positive. This is important, for example, in medical diagnoses.

Example 2 Suppose each tuple is a diagnostic test that looks for a positive reaction helping differentiate characteristics of similar microbiologic genera. The tests performed are Gelatinase, Mannitol, Inositol, OF(glucose) (a ‘+’ means a positive test) and whether the organism is toxic (a ‘+’ means the organism is toxic.) On the left is the instance. On the right, rules 1. and 2. are classical association rules, while rules 3. through 5. relate negative as well as positive characteristics.

test-id	Gel	Man	Ino	OF-glu	Toxic		
2	+	+	-	+	+	$\text{Gel} \wedge \text{OF-glu}$	$\Rightarrow \text{Toxic}$
92	+	+	-	+	+	Man	$\Rightarrow \text{OF-glu}$
33	-	-	+	-	-	Man_+	$\Rightarrow \text{Ino}_-$
54	+	-	+	-	-	Man_-	$\Rightarrow \text{Ino}_+$
45	+	-	+	+	+	OF-glu_-	$\Rightarrow \text{Toxic}_-$

■

Associations actually began with a very different representation. The characterization of ARs did not begin with the representation used above, although it is the current vogue, but with a form closer to the typical data representation. An example in the medical context begins with the *patient-symptom* relation, denoted PS, with attributes *pid* and *symptom*. This data indicates the *symptom weakness* always occurs in the presence of *symptom fatigue*, symbolized as *fatigue* $\Rightarrow$ *weakness*. Then $y \Rightarrow z$ iff $(\forall p \in \text{PS. pid})[(p, y) \in$

*This is called *genericity* in database query theory

$PS \Rightarrow \langle p, z \rangle \in PS$. To express this in the CE framework, define the estimator $\mathcal{E} = [\text{symptom}]$ and classifier $\mathcal{C} = \{C_y, C_y^-\}_{y \in \text{symptom}}$, where

$$C_y = \{\langle p, y \rangle \mid \langle p, y \rangle \in PS \wedge \langle p, \text{weakness} \rangle \in PS\}. \quad (4)$$

$$C_y^- = \{\langle p, y \rangle \mid \langle p, y \rangle \in PS \wedge \langle p, \text{weakness} \rangle \notin PS\}. \quad (5)$$

This classifier/estimator pair in fact facilitates the determination of all y such that $y \Rightarrow \text{weakness}$. To see this, observe that $E_y = C_y \cup C_y^-$. $C_y^- \neq \emptyset$ iff there is some pid p such that $\langle p, y \rangle \in PS$ but $\langle p, \text{weakness} \rangle \notin PS$. So, $y \Rightarrow \text{weakness}$ iff $[\text{symptom}](y) = C_y$.

Example 3 Suppose PS exists as the instance on the left. The CE is on the right. Observe that this instance exhibits $\text{fatigue} \Rightarrow \text{weakness}$ and the trivial $\text{weakness} \Rightarrow \text{weakness}$ as perfect associations. The association $\text{cough} \Rightarrow \text{weakness}$, however, fails to hold since the estimator class, $[\text{symptom}](\text{cough}) = \{t_5, t_6\}$, is split into two in the classifier.

	pid	symptom	
t_1	1	<i>fatigue</i>	
t_2	1	<i>weakness</i>	
t_3	2	<i>fatigue</i>	$\mathcal{E} = \{\{t_1, t_3\}, \{t_2, t_4\}, \{t_5, t_6\}\}$
t_4	2	<i>weakness</i>	$\mathcal{C} = \{\{t_1, t_3\}, \{t_2, t_4\}, \{t_5\}, \{t_6\}\}$
t_5	2	<i>cough</i>	
t_6	3	<i>cough</i>	

■

Of course with fixed y and z , $y \Rightarrow z$ can be transformed into an AR, viz., $\text{symptom} = y \Rightarrow \text{symptom} = z$. But this transformation must create a new Boolean attribute for each value of symptom. Except by explicit search, ARs cannot find all z such that $\text{symptom} \Rightarrow z$, much less the even more general question considered next. On the other hand, exploration of associations involving several attributes requires quantification over many variables or, equivalently, joining the relation with itself, perhaps several times.

Consideration of the more general question of discovering all y, z pairs such that $y \Rightarrow z$ shows the first example where the base relation does not have enough “room” for relevant classifier and estimator partitions.* It is necessary to build a larger relation, in this case $\mathcal{U}$ is $PS \times PS \cdot \text{symptom}$. Then the partitions are $C_z = \{t \mid t = \langle p, y, z \rangle\}$ and $E_y = \{t \mid t = \langle p, y, z \rangle\}$. Thus E_y straddles C_z iff it is not the case that $y \Rightarrow z$. This example also clearly shows that the CE framework is a conceptual tool and not a recipe for implementation, since a single pass through the data will suffice to compute the associative metrics

*This is an “input/output complexity” issue.

provided that **PS.symptom** is of moderate size. Again, exploration involving several attributes requires wide tuples.

6 METRICS

There is a large body of work on metrics for data mining (Silberschatz and Tuzhilin 1995, Fayyad and G.Piatetsky-Shapiro 1996b, Fayyad and G.Piatetsky-Shapiro 1996c, J.Kivinen and Mannila 1995, Mannila 1996). In the CE framework, the goodness of an estimator can be decided either *quantitatively* or *qualitatively*. The CE framework does not itself prefer one metric over another, but provides a way to describe and evaluate them.

In both kinds of goodness measure, there is the dimension of breadth and specificity that enables us to focus the CE to our need. This leads us to an equivalent analytic definition of indeterminate set,

$$\mathfrak{I}_C^\mathcal{E} = \bigcup \{E \cap C \mid E \cap C \neq E, \text{ for } E \in \mathcal{E}, C \in \mathcal{C}\}. \quad (6)$$

Proposition 7 For two partitions $\mathcal{E}, \mathcal{C}$ over $\mathcal{U}$, $\mathcal{I}_C^\mathcal{E} = \mathfrak{I}_C^\mathcal{E}$

Proof Suppose there is a straddler E in the union forming $\mathcal{I}_C^\mathcal{E}$. Then let $C_1, \dots, C_k$, be classifier sets that E straddles and $E = \bigcup E \cap C_i$. Since $E \not\subseteq C_i$, each $E \cap C_i$ is in the union defining $\mathfrak{I}_C^\mathcal{E}$ and thus, $\mathfrak{I}_C^\mathcal{E}$ contains the straddler E . Now, suppose there is an S in union forming $\mathcal{I}_C^\mathcal{E}$. Then $S \subseteq E$, where $E \in \mathcal{E}$ straddles some $C \in \mathcal{C}$. Hence, $\mathcal{I}_C^\mathcal{E}$ contains the straddler E and also S . ■

Observe that like $\mathcal{I}_C^\mathcal{E}$, $\mathfrak{I}_C^\mathcal{E}$ is also well-defined for two collections of sets $\mathcal{E}, \mathcal{C}$. This allows us to examine $\mathcal{E}$ one column at a time and $\mathcal{C}$ one row at a time, viz., $\mathfrak{I}_C^\mathcal{E} = \bigcup_{E \in \mathcal{E}} \mathfrak{I}_C^{\{E\}} = \bigcup_{C \in \mathcal{C}} \mathfrak{I}_{\{C\}}^\mathcal{E}$. From now on, we write $\mathfrak{I}_C^E$ and $\mathfrak{I}_C^\mathcal{E}$, understanding they stand for $\mathfrak{I}_C^{\{E\}}$ and $\mathfrak{I}_{\{C\}}^\mathcal{E}$, respectively. Note that $\mathcal{I}_C^E$ is more robust than $\mathcal{I}_C^\mathcal{E}$, since $\mathfrak{I}_C^E = \mathcal{I}_C^E$, but $\mathfrak{I}_C^\mathcal{E} \subsetneq \mathcal{I}_C^\mathcal{E}$ unless $\mathcal{I}_C^\mathcal{E} = \emptyset$.

Given an estimator $\mathcal{E} = \{E_1, \dots, E_k\}$ and classifier $\mathcal{C} = \{C_1, \dots, C_n\}$, there is a multitude of possible measures. Some of those that suggest themselves are

1. Indeterminate count of P : $\mathbf{c}_\mathcal{E}(C) = |\mathfrak{I}_C^\mathcal{E}|$. Somewhat local, made wrt a particular $\mathcal{C}$ -class C .
2. Indeterminate count of E : $\mathbf{c}_\mathcal{C}(E) = |\mathfrak{I}_C^E|$. Somewhat local, made wrt a particular $\mathcal{E}$ -class E .
3. Indeterminate count at E, C : $|\mathfrak{I}_C^E|$. Very local, made wrt particular $\mathcal{C}$ -class and $\mathcal{E}$ -class.
4. Subtotal indeterminate count $\mathcal{E}' \subseteq \mathcal{E}, \mathcal{C}' \subseteq \mathcal{C}$: $C(\mathcal{E}', \mathcal{C}') = \sum_{C \in \mathcal{C}'} \mathbf{c}_{\mathcal{E}'}(C) = \sum_{E \in \mathcal{E}'} \mathbf{c}_{\mathcal{C}'}(E)$.
5. Determinate precision of C : $d_\mathcal{E}(C) = \frac{|C| - \mathbf{c}_\mathcal{E}(C)}{|C|}$.

6. Indeterminate precision of P : $i_{\mathcal{E}}(C) = \frac{|C|}{|c| + \mathbf{c}_{\mathcal{E}}(C)}$.
7. Normalized determinate precision of P : $\mathbf{d}_{\mathcal{E}}(C) = \frac{|C| - \mathbf{c}_{\mathcal{E}}(C)}{|U|}$.
8. Total precision: $\mathbf{C}(\mathcal{E}, C) = \sum_{C \in \mathcal{P}} \mathbf{d}(C)$.

Each of these metrics can be used to evaluate which of these two estimators is *quantitatively better*. An estimator $\mathcal{E}$ is *qualitatively better* than estimator $\mathcal{F}$ iff $\mathcal{I}_{\mathcal{E}}^{\mathcal{E}} \subseteq \mathcal{I}_{\mathcal{F}}^{\mathcal{F}}$, in other words, $\mathcal{F} \preceq \mathcal{E}$.

Example 4 (Imperfect ARs) Since we already have a *perfect* characterization of ARs, we can relax the confidence to describe ARs that are less than perfect. For this example we will use the following relation $\mathbf{r}$ (left)

$\mathbf{r}$	A	B	C	D
t_1	+	+	-	+
t_2	+	-	-	+
t_3	-	+	+	+
t_4	+	+	-	+
t_5	+	+	-	+
t_6	-	-	-	+
t_7	+	+	-	-

Say we are interested in $B \wedge D \Rightarrow A$. Then $\mathcal{E} = \{\{t_1, t_3, t_4, t_5\}, \{t_2, t_6, t_7\}\}$ and $\mathcal{C} = \{\{t_1, t_4, t_5\}, \{t_2, t_3, t_6, t_7\}\}$. The support is $|\mathfrak{B}_{BDA+}^{BD+}|/|\mathbf{r}| = 43\%$ and the confidence is $|BDA+|/|\mathfrak{B}_{BDA+}^{BD+}| = 75\%$. ■

7 SUMMARY CONCLUSIONS

This paper has presented a framework which

- unifies a variety of data mining and database concepts.
- establishes both a perfect condition and approximate condition.
- provides a range of breadth/specificity so important to drill-down data mining.
- supports a wide variety of quantitative and qualitative metrics

Furthermore, many of the CE constructions are surprisingly simple. It is the fact that such a variety of topics are handled simply makes this framework significant.

We began the paper by drawing parallels with relation database theory. Continuing this metaphor, we feel we still need to discover the analogs of SQL, QBE, efficient join algorithms, query optimization strategies, *etc.* In

addition, we intend to “push the envelope” of the CE framework in such areas as temporal information, nested relations, and sampling.

One final example suggests a broad range of possibilities. The general issue involves relationships between the values of attributes. Consider a relation over X, Y, Z . We are interested in all values $t.x$ such that $t.Y > t.Z$. This is in fact an instance of a property, in that the classifier is merely a partition distinguishing those t for which $t.Y > t.Z$ holds from those which it does not. Said another way, the specified problem is a kind of functional dependency, except that the dependent is derived rather than base data. The generalization is immediate; by using functions of the attribute values rather than just the values themselves to define the equivalence classes, a huge variety of data mining is possible.

REFERENCES

- Abiteboul, S., Hull, R. and Vianu, V.: 1995a, *Foundation of Databases*, Addison-Wesley Publishing Company, Reading, MA.
- Abiteboul, S., Hull, R. and Vianu, V.: 1995b, *Foundation of Databases*, Addison-Wesley Publishing Company, Reading, MA., pp. 166–169.
- Agrawal, R., Imielinski, T. and A.Swami: 1993, Mining association rules between sets of items in large databases, *Proc. of ACM SIGMOD Conference on Management of Data (SIGMOD'93)*, pp. 207–216.
- Agrawal, R. and R.Srikant: 1994, Fast algorithms for mining association rules in large databases, *Proc. of the 20th International Conference on Very Large Data Bases*, pp. 478–479.
- Armstrong, W.: 1974, Dependency structures of data base relationships, *Proc. IFIP Congress*, North Holland, Amsterdam, The Netherlands, pp. 580–583.
- Beeri, C., R.Fagin and J.H.Howard: 1977, A complete axiomatization for functional and multivalued dependencies, *Proc. Second ACM Symp. on Principles of Database Systems*, pp. 45–62.
- Codd, E.: 1970, A relational model of data for large shared data banks, *Communications of the ACM* 13(6), 377–387.
- Codd, E.: 1972, Relational completeness of database sublanguages, in R. Rustin (ed.), *Courant Computer Science Symposium 6: Data Base Systems*, Prentice-Hall, Englewood Cliffs, NJ, pp. 33–64.
- Delobel, C.: 1978, Normalization and hierarchical dependencies in the relational model, *ACM Trans. on Database Systems* 3(3), 201–222.
- Fagin, R.: 1977, Multivalued dependencies and a new normal form for relational databases, *ACM Trans. on Database Systems* 2(3), 262–278.
- Fayyad, U. and G.Piatetsky-Shapiro (eds): 1996a, *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA., pp. 1–34.
- Fayyad, U. and G.Piatetsky-Shapiro (eds): 1996b, *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA., pp. 1–34.

- Fayyad, U. and G.Piatetsky-Shapiro (eds): 1996c, *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA.
- Harinarayan, V., J.D.Ullman and Rajaraman, R.: 1996, Implementing data cubes effectively, *Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data*, pp. 205–216.
- Imielinski, T. and H.Mannila: 1996, A database perspective on knowledge discovery, *Communications of the ACM* **39**(11), 58–64.
- J.Kivinen and Mannila, H.: 1995, Approximate dependency inference from relations.
- Lin, T. (ed.): 1997, *The Fifth International Workshop on Rough Sets and Soft Computing (RSSC'97)*.
- Mannila, H.: 1996, Data mining: machine learning, statistics, and databases, *Proc. of the 8 International Conference on Scientific and Statistical Database Management*, pp. 1–6.
- Mannila, H.: 1997, Methods and problems in data mining, in F. Afrati and P.Kolaitis (eds), (to appear) *Proc. of International Conference on Database Theory, Delphi, Greece*, Springer-Verlag.
- Mannila, H., H.Toivonen and A.Inkeri Verkamo: 1994, Efficient algorithms for discovering association rules, *Proc. AAAI Workshop on Knowledge Discovery in Databases*, pp. 181–192.
- Pawlak, Z.: 1982, Rough sets, *International Journal of Computer and Information Sciences* **11**(5), 341–356.
- Pawlak, Z.: 1991, *Rough Sets: Theoretical Aspects of Reasoning About Data*, Kluwer Academic Publishers, The Netherlands.
- Pawlak, Z., Grzymala-Busse, J., Slowinski, R. and Ziarko, W.: 1995, Rough sets, *Communications of the ACM* **38**(11), 88–95.
- Silberschatz, A. and Tuzhilin, A.: 1995, On subjective measure of interestingness in knowledge discovery, *Proc. 1st International Conference on Knowledge Discovery and Data Mining (KDD'95) (Montreal, Canada)*, AAAI Press, Menlo Park, CA., pp. 275–281.
- Simoudis, E.: 1996, Reality check for data mining.
- Slowinski, R. and Stefanowski, J.: 1996, Rough-set reasoning about uncertain data, *Fundamenta Informaticae* **27**(2/3), 229–243.
- Ullman, J.: 1988a, *Principles of Database and Knowledge-base Systems: Volume I: Classical Database Systems*, Computer Science Press, Rockville, MD.
- Ullman, J.: 1988b, *Principles of Database and Knowledge-base Systems: Volume I: Classical Database Systems*, Computer Science Press, Rockville, MD., pp. 413–420.
- Vardi, M.: 1987, Fundamentals of dependency theory, in E. Borger (ed.), *Trends in Theoretical Computer Science*, Computer Science Press, Rockville, MD, pp. 171–224.

BIOGRAPHY

Mehmet Dalkilic is currently completing his dissertation in Computer Science at Indiana University. He received a BA in Chemistry from Indiana University (1984) and initially participated in the MD/PhD program (Biochemistry) at Indiana University/Purdue University at Indianapolis. His research interests include database systems, chemical information systems, software design and implementation, and data mining.

Edward Robertson received the BS in Mathematics from Cal. Tech. in 1966 and the PhD in Computer Science from the University of Wisconsin in 1970. He is currently Professor of Computer Science at Indiana University, Bloomington. His research interests include theory of computation, database systems, and computer systems design and implementation, especially software engineering. He has served on the faculty of the University of Ghana, the University of Waterloo, Penn State University, and the University of Nairobi as a Fulbright Fellow.

Dirk Van Gucht is currently Associate Professor of Computer Science at Indiana University, Bloomington. He received a Licentiate in Mathematics from Vrije Universiteit Brussel (1982) and a PhD in Computer Science from Vanderbilt University (1984). He was a visiting professor at the University of Limburg and the University of Antwerp in 1994. Van Gucht's research deals with the design and implementation of database models and programming languages for database applications that use information objects with a complex external and internal organization.