Technical Report No. 480

CE: The Classifier–Estimator Framework for Data mining

Mehmet M. Dalkilic Edward L. Robertson Dirk Van Gucht

Computer Science, Indiana University,

Bloomington, IN 47405, USA.

Email: {dalkilic, edrbtsn, vgucht}@cs.indiana.edu

Abstract

The aim of this research is to establish a coherent framework for data mining in the relational model. 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 relational instance or may involve an "oracle" beyond the extant data. The estimator is typically simply expressible using the relational 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.

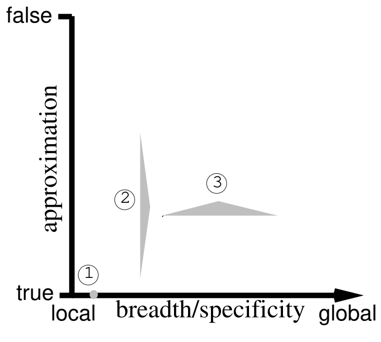
1 Introduction

Data mining faces a predicament similar to that databases faced prior to Codd's [8] 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 [16, 20]. 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 the *relational model* [8, 1, 27]. Our exemplar is the success of the relational model in addressing two particular problems 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 [12]. This "drawing forth" is made *wrt* some *concept*[1] that varies along two dimensions. First, there is a degree of breadth/specificity, in that the mining may look for general aspects of all the data or specific details of a select subset (variations of which are commonly refered to as "roll-up" and "drill-down" in the *data cube* approach [15]). Second, there is a degree of approximate validity, in that the mined information lies somewhere between true and false

(See Fig. 1.) 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 Ys 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

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



The three shaded regions, marked by 1,2, and 3, represents some play on these dimensions for a particular concept of interest. Both "local" and "global" are relative terms, but indicate a kind of conceptual gradient from individual instances, to sets of instances that share some properties, and so forth. 1 signifies a perfectly true, mostly local occurrence of the concept. 2—more global than 1—allows the concept to be approximately true. 3, on the other hand, has a narrow interpretation of approximation, while looking for a broader interpretation of the concept.

Figure 1: The two dimensions of data mining: breadth/specification and approximation.

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 concept, functional dependency. Section 3 and 4 examine a popular data mining framework, rough sets, and begin to formally develop our framework. 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 Turning a Database Concept into a Data mining Tool

This section provides an intuitive tour of the Classifier/Estimator (CE) framework using a well–known concept from database theory: functional dependency (FD) [9, 30, 28, 2]. Intuitively, the notion of functional dependency says that each value of one attribute always implies one specific value of another attribute[2]; thus, the word "functional" is applied because the second attribute is a function of the first. An example of a functional dependency in a medical care information system is that `benefit_code` functionally determines `deductible`. (This also illustrates that functional dependencies typically come from "business rules" which provide the absolute certainty of a perfect fact.)

Formally, let $\mathbf{r}$ be a relation instance over attributes $X, Y, \ldots$. For a tuple $t \in \mathbf{r}$, $t.X$ denotes the X component value of t. Similarly, $\mathbf{r}.X$ denotes the set of all X values of tuples in $\mathbf{r}$. With this notation, the functional dependence of Y on X, also written "X functionally determines Y" or "$X \rightarrow Y$", is defined to hold iff $\phi(X,Y)$, where $\phi(X,Y) \equiv (\forall x \in \mathbf{r}.X)(\exists y \in \mathbf{r}.Y)(\forall t \in \mathbf{r})[t.X = x \Rightarrow t.Y = y]$. Using the familiar semantics of first order logic (*FOL*), ϕ is perfect in that its value is either true or false.

We now describe an informal (the formalism follows later) characterization of $X \rightarrow Y$ in the CE framework. Given an $\mathbf{r}$ for which we wish to test $X \rightarrow Y$, we first define two classes of parameterized sets: $E_x = \{t \in \mathbf{r} | t.X = x\}$ and $C_y = \{t \in \mathbf{r} | t.Y = y\}$. It is obvious that distinct x values give distinct and disjoint E_x and that each $t \in \mathbf{r}$ belongs to E_x for that x such that $t.X = x$. Thus, $\{E_x\}_{x \in \mathbf{r}.X}$ is a partition of $\mathbf{r}$. Similarly, $\{C_y\}_{y \in \mathbf{r}.Y}$ is a partition of $\mathbf{r}$. The stage is now set, since the CE framework requires two partitions, the *classification* and the *estimation*–in this case, $\{C_y\}_{y \in \mathbf{r}.Y}$ and $\{E_x\}_{x \in \mathbf{r}.X}$, respectively. Then $\phi(X,Y)$ holds iff every E_x is a subset of one specific C_y. Writing this symbolically, including an explicit formulation of subset, gives exactly the same form as our initial sentence, *viz.*, $(\forall E_x)(\exists\, C_y)(\forall t \in \mathbf{r})[t \in E_x \Rightarrow t \in C_y]$ or equivalently, $(\forall E_x)(\exists\, C_y)[\, E_x \subseteq C_y\,]$. This alternative formulation provides nothing to standard functional dependency theory (beyond demonstrating the adequacy of the CE framework to capture this important

[2] We use individual attributes here in order to simplify notation. Later we give more complete coverage of functional dependencies.

concept), but the classifier/estimator pair is the central feature of data mining to which we now return.

It would be highly significant to discover that $X \rightarrow Y$, but the likelihood of newly discovering this perfect fact is so low that we would probably never try—our resources are better used in discovering something else, something *close* to the truth. One way in which $X \rightarrow Y$ may be investigated is to look at its subclasses, the simplest of which is to see whether a dependency holds *locally* for specific values of x and y. We write this local dependency as $x \stackrel{XY}{\rightarrow} y$, or simply $x \rightarrow y$ when X and Y are clear from the context, defined as $(\forall t \in \mathbf{r})[t.X = x \Rightarrow t.Y = y]$ and witnessed by $E_x \subseteq C_y$. In the medical example, this would be a statement that a certain symptom always determines a unique disease; oncologists know that the presence of alpha–fetoprotein always indicates liver cancer. The evidence for this would be that $E_{\texttt{alpha-fp}} \subseteq C_{\texttt{liver-cancer}}$. At a more global level, it might be interesting to find all X such that $\phi(X, \mathbf{Y})$ for some $\mathbf{Y}$.

Another dimension in which FDs can be investigated is to see how well this fits the data (this is related to [17]). This issue is 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 judgement of the quality of $X \rightarrow Y$ is that problems 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. 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. Another approximation criterion is local, finding all x such that 90% of E_x belongs to the corresponding C_y. If the percentage parameter is less than 50%, this could yield an approximation to a functional dependency which was no longer functional, in the sense that $x \rightsquigarrow y_1$ and $x \rightsquigarrow y_2$, for distinct y_i. Even though the presence `acid-phosphatase` does not always indicate a unique disease, it is still highly significant in that it either indicates `prostate-cancer` or some particular kind of `leukemia`, each with roughly 45% of the cases. Of course, this theme could be elaborated at great length, always considering $E_x \cap C_y$ and $E_x \cap \bar{C}_y$, where both are non–empty.

3 From *Rough Sets* to Classifier/Estimator

Rough sets have become a popular framework for data mining investigations[24, 26, 18], but they cannot capture many important database concepts that also are relevant 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 [22, 23] 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 that must hold for at least one member of the entire class (possibility) or 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:

$$\begin{array}{ll} \text{lower–bound} & {}_{\wedge}P_{\mathcal{E}} = \bigcup\{E | E \subseteq P, \text{ for } E \in \mathcal{E}\} \\ \text{upper–bound} & {}_{\vee}P_{\mathcal{E}} = \bigcup\{E | E \cap P \neq \emptyset, \text{ for } E \in \mathcal{E}\} \end{array}$$

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

1. Analyzing a property P by investigating its relationship(s) with certain partitions of $\mathcal{U}$.
2. Evaluating the interaction of the property P and the partition $\mathcal{E}$ at the level of individual classes of $\mathcal{E}$.
3. Measuring 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 (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 framework 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 $x\mathcal{B}y$ to indicate that x and y belongs 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 3.1 For $E \in \mathcal{E}$ and $C \in \mathcal{C}$, E *straddles* C, written $E \between C$, 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 3.2 The *indeterminate set*, written $\mathcal{I}^{\mathcal{E}}_{\mathcal{C}}$, is $\mathcal{I}^{\mathcal{E}}_{\mathcal{C}} = \bigcup\{E | E \text{ is a straddler}\}$. ■

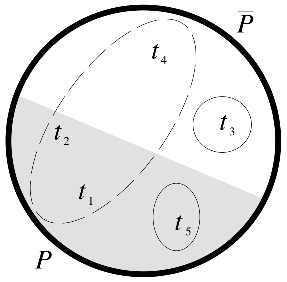
Proposition 3.1 ${}_{\wedge}P_{\mathcal{E}} = P - \mathcal{I}^{\mathcal{E}}_{\mathcal{P}}$.

Proof
$$\begin{array}{rcl} P - \mathcal{I}^{\mathcal{E}}_{\mathcal{P}} & = & \bigcup(E \cap P) - \mathcal{I}^{\mathcal{E}}_{\mathcal{P}} \\ & = & \bigcup\{E | E \subseteq P\} \,\cup\, \bigcup\{E | E \between P\} - \bigcup\{E | E \between P\} \\ & = & \bigcup\{E | E \subseteq P\} \end{array}$$
■

Proposition 3.2 ${}_{\vee}P_{\mathcal{E}} = P \cup \mathcal{I}^{\mathcal{E}}_{\mathcal{P}}$.

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

Example 3.1 (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 $\mathcal{I}^{\mathcal{E}}_{\mathcal{P}} = \{t_1, t_2, t_4\}$ and the lower and upper approximations of P are ${}_{\wedge}P_{\mathcal{E}} = P - \mathcal{I}^{\mathcal{E}}_{\mathcal{P}} = \{t_5\}$ and ${}_{\vee}P_{\mathcal{E}} = P \cup \mathcal{I}^{\mathcal{E}}_{\mathcal{P}} = \{t_1, t_2, t_4, t_5\}$.



$\mathcal{U}$ 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 $\mathcal{I}^{\mathcal{E}}_{\mathcal{C}} = \emptyset$. We know this means there are no straddlers, but there is another meaning we can associate with this condition, *refinement.* Given two partitions $\mathcal{A}, \mathcal{B}$ over $\mathcal{U}$, $\mathcal{B}$ is a *refinement* of $\mathcal{A}$, written $\mathcal{A} \preccurlyeq \mathcal{B}$, 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 4.1 $\mathcal{I}^{\mathcal{E}}_{\mathcal{C}} = \emptyset$ iff $\mathcal{C} \preccurlyeq \mathcal{E}$.

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

Consider now when $\mathcal{I}_{\mathcal{C}}^{\mathcal{E}} \neq \emptyset$. Although this means that $\mathcal{C} \preccurlyeq \mathcal{E}$ does not hold, we can associate some measure of "mismatch" wrt 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 much more robust 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, we use the following notations, $R[A]$ is a relational schema where $A = \{A_1, \ldots, A_n\}$ and we assume an instance $\mathbf{r}$ 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 measure quality in 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*[6, 3]), 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, 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 5.1 $X \rightarrow Y$ 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 $E \subseteq C$, and E cannot straddle C. But E cannot straddle any other $C' \in [Y]$, because $E \cap C' = \emptyset$. 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 $C' \in [P]$. 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)[10, 11, 29], 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[7, 29]. 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 5.1 *X multidetermines Y in the context Z* (mention of Z is often ommited 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 $\mathcal{C}$ such that $s\mathcal{C}t \equiv N_y(s) = N_y(t)$ and $\mathcal{E} = [X]$, the desired result is immediate.

Proposition 5.2 *$X \twoheadrightarrow Y|Z$ iff $C \preccurlyeq E$.* ■

5.3 Association Rules

Association rules (AR) have been gaining popularity in both data mining and databases, discussed in [4, 5, 21, 20]. As a concept, AR begins 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 $\{t | t \in \mathbf{r} \ \wedge \ t.A_i = +, \text{ for each } A_i \in W\}$. Without loss of generality, let $X = \{A_1, \ldots, A_k\}$ and $Y = \{A_{k+1}, \ldots, A_m\}$. Expression $A_1 \ \wedge \ldots \wedge A_k \Rightarrow A_{k+1} \ \wedge \ldots \wedge A_m$, 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 are "value–blind"[3], 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 $\mathcal{E} = \{X_+, \overline{X}_+\}$.

[3] This is called *genericity* in database query theory

Proposition 5.3 $X \Rightarrow Y, c = 1$ iff $\mathcal{C} \preccurlyeq \mathcal{E}$.

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 $\mathcal{C} \preccurlyeq \mathcal{E}$. Since $t_+ \in XY_+ \cap X_+$, this can only hold if $XY_+ = X_+$. This gives $c = 1$ as required. ■

An AR is actually on a local fact related to an FD. That is, the above example is $+, \ldots, + \overset{X,Y}{\rightarrow} +, \ldots, +$. 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 5.1 Suppose each tuple is a dianostic 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	+	+	-	+	+
92	+	+	-	+	+
33	-	-	+	-	-
54	+	-	+	-	-
45	+	-	+	+	+

$$
\begin{aligned}
\texttt{Gel} \wedge \texttt{OF-glu} &\Rightarrow \texttt{Toxic} \\
\texttt{Man} &\Rightarrow \texttt{OF-glu} \\
\texttt{Man}_+ &\Rightarrow \texttt{Ino}_- \\
\texttt{Man}_- &\Rightarrow \texttt{Ino}_+ \\
\texttt{OF-glu\_} &\Rightarrow \texttt{Toxic}_-
\end{aligned}
$$

■

There is another variety of "associations" that begins with a very different 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* $\rightrightarrows$ *weakness*. Then $x \rightrightarrows y$ iff $(\forall p \in \texttt{PS.pid})[\langle p, x\rangle \in \texttt{PS} \Rightarrow \langle p, y\rangle \in \texttt{PS}]$. To express this in the CE framework, define the classifier $\mathcal{C} = [\texttt{symptom}]$ and estimator $\mathcal{E} = \{E_y, E_y^-\}_{y \in \texttt{symptom}}$ where

$$
\begin{aligned}
E_y &= \{t | t \in \texttt{PS} \ \wedge \ t.\texttt{symptom} = y \ \wedge \ (\exists t' \in \texttt{PS})[t.\texttt{pid} = t'.\texttt{pid} \ \wedge \ t'.\texttt{symptom} = \textit{fatigue}] \\
E_y^- &= \{t | t \in \texttt{PS} \ \wedge \ t.\texttt{symptom} = y \ \wedge \ (\nexists t' \in \texttt{PS})[t.\texttt{pid} = t'.\texttt{pid} \ \wedge \ t'.\texttt{symptom} = \textit{fatigue}]
\end{aligned}
$$

This classifier/estimator pair in fact facilitates the determination of all y such that *fatigue* $\rightrightarrows y$. To see this, observe that `symptom` $y = E_y \cup E_y^-$. $E_y^- \neq \emptyset$ iff there is some `pid` p such that $\langle p, \textit{fatigue}\rangle \in \texttt{PS}$ but $\langle p, y\rangle \notin$ PS. So, *fatigue* $\rightrightarrows y$ iff `symptom` $y = E_y$.

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

Consideration of the more general question of discovering all x, y pairs such that $x \rightrightarrows y$ shows the first example where the base relation does not have enough "room" for relevant classifier and estimator partitions.[4] It is necessary to build a larger relation, in this case $\mathcal{U}$ is $\texttt{PS} \times \texttt{PS.symptom}$. Then the partitions are $C_y = \{t | t = \langle p, x, y \rangle\}$ and $E_x = \{t | t = \langle p, x, y \rangle\}$. Thus E_x straddles C_y iff it is not the case that $x \rightrightarrows y$. This example also clearly shows that the CE framework is a conceptual tool and not a recipe for implementation, since in most cases, a single pass through the data will suffice to compute the associative metrics.

6 Metrics

There is a large body of work on metrics [25, 13, 14, 17, 19]. 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/specificity that enables us to focus the CE to our need. This leads us to an equivalent analytic definition of indeterminate set,
$\mathfrak{I}_{\mathcal{C}}^{\mathcal{E}} = \bigcup \{E \cap C |\ E \cap C \neq E,\ \text{for } E \in \mathcal{E}, C \in \mathcal{C}\}$.

Proposition 6.1 For two partitions $\mathcal{E}, \mathcal{C}$ over $\mathcal{U}$, $\mathcal{I}_{\mathcal{C}}^{\mathcal{E}} = \mathfrak{I}_{\mathcal{C}}^{\mathcal{E}}$

Proof Suppose there is a straddler E in the union forming $\mathcal{I}_{\mathcal{C}}^{\mathcal{E}}$. Then let $C_1, \ldots, C_k$, be classifier sets that E straddles and $E = \bigcup E \cap C_i$. Since $E \between C_i$, each $E \cap C_i$ is in the union defining $\mathfrak{I}_{\mathcal{C}}^{\mathcal{E}}$ and thus, contains the straddler E. Now, suppose there is an S in union forming $\mathfrak{I}_{\mathcal{C}}^{\mathcal{E}}$. Then $S \subseteq E$, where $E \in \mathcal{E}$ straddles some $C \in \mathcal{C}$. Hence, $\mathcal{I}_{\mathcal{C}}^{\mathcal{E}}$ contains the straddler E and also S. ■

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

Given an estimator $\mathcal{E} = \{E_1, \ldots, E_k\}$ and classifier $\mathcal{C} = \{C_1, \ldots, C_n\}$, there is a multitude of possible measures. Some of these that suggests themselves to us 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.

2. Indeterminate count of E: $\mathbf{c}_{\mathcal{C}}(E) = |\,\mathfrak{I}_{\mathcal{C}}^{E}\,|$. Somewhat local, made wrt a particular $\mathcal{E}$–class.

3. Indeterminate count at E, C: $|\mathfrak{I}_{C}^{E}|$. Very local, made wrt a particular $\mathcal{C}$–class, $\mathcal{E}$–class.

4. Subtotal indeterminate count $\mathcal{E}' \subseteq \mathcal{E}$, $\mathcal{C}' \subseteq \mathcal{C}$: $C(\mathcal{E}', \mathcal{P}') = \sum_{C \in \mathcal{C}'} \mathbf{c}_{\mathcal{E}'}(C) = \sum_{E \in \mathcal{E}'} \mathbf{c}_{\mathcal{C}'}(E)$.

5. Determinate precision of C: $d_{\mathcal{E}}(C) = \frac{|C|\ -\ \mathbf{c}_{\mathcal{E}}(C)}{|C|}$.

[4] This is an "input/output complexity" issue.

6. Indeterminate precision of P: $i_{\mathcal{E}}(C) = \frac{|C|}{|c| + \mathbf{c}_{\mathcal{E}}(C)}$.

7. Normalized determinate precision of P: $\mathbf{d}_{\mathcal{E}}(C) = \frac{|C| - \mathbf{c}_{\mathcal{E}}(c)}{|\mathcal{U}|}$.

8. Total precision: $\mathbf{C}(\mathcal{E}, \mathcal{C}) = \sum_{C \in \mathcal{P}} \mathbf{d}(C)$

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 $\mathcal{I}_{\mathcal{C}}^{\mathcal{E}} \subseteq \mathcal{I}_{\mathcal{C}}^{\mathcal{F}}$, in other words, $\mathcal{F} \preccurlyeq \mathcal{E}$.

7 Summary Conclusions

This paper has presented a framework which

- unifies a variety of data mining and atabase concepts.
- 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 paralles 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 function of the attribute values rather than just the values themselves to define the equivalence classes, a huge variety of data mining is possible.

References

[1] Abiteboul, S., Hull, R., and Vianu, V. *Foundation of Databases.* Addison-Wesley Publishing Company, Reading, MA., 1995.

[2] Abiteboul, S., Hull, R., and Vianu, V. *Foundation of Databases.* Addison-Wesley Publishing Company, Reading, MA., 1995, pp. 139–259.

[3] Abiteboul, S., Hull, R., and Vianu, V. *gFoundation of Databases.* Addison-Wesley Publishing Company, Reading, MA., 1995, pp. 166–169.

[4] Agrawal, R., Imielinkski, T., and A.Swami. Mining association rules between sets of items in large databases. In *Proc. of ACM SIGMOD Conference on Management of Data (SIGMOD'93)* (May 1993), pp. 207–216.

[5] Agrawal, R., and R.Srikant. Fast algorithms for mining association rules in large databases. In *Proc. of the 20th International Conference on Very Large Data Bases* (September 1994), pp. 478–479.

[6] Armstrong, W. Dependency structures of data base relationships. In *Proc. IFIP Congress* (Amsterdam, The Netherlands, 1974), North Holland, pp. 580–583.

[7] Beeri, C., R.Fagin, and J.H.Howard. A complete axiomatization for functional and multivalued dependencies. In *Proc. Second ACM Symp. on Principles of Database Systems* (1977), pp. 45–62.

[8] Codd, E. A relational model of data for large shared data banks. *Communications of the ACM 13*, 6 (1970), 377–387.

[9] Codd, E. Relational completeness of database sublanguages. In *Courant Computer Science Symposium 6: Data Base Systems* (Englewood Cliffs, NJ, 1972), R. Rustin, Ed., Prentice-Hall, pp. 33–64.

[10] Delobel, C. Normalization and hierarchical dependencies in the relational model. *ACM Trans. on Database Systems 3*, 3 (1978), 201–222.

[11] Fagin, R. Multivalued dependencies and a new normal form for relational databases. *ACM Trans. on Database Systems 2*, 3 (1977), 262–278.

[12] Fayyad, U., and G.Piatetsky-Shapiro, Eds. *Advances in Knowledge Discovery and Data Mining.* AAAI Press, Menlo Park, CA., 1996, pp. 1–34.

[13] Fayyad, U., and G.Piatetsky-Shapiro, Eds. *Advances in Knowledge Discovery and Data Mining.* AAAI Press, Menlo Park, CA., 1996.

[14] Fayyad, U., and G.Piatetsky-Shapiro, Eds. *Advances in Knowledge Discovery and Data Mining.* AAAI Press, Menlo Park, CA., 1996.

[15] Harinarayan, V., J.D.Ullman, and Rajaraman, R. Implementing data cubes effectively. In *Proc. 1996 ACM-SIGMOD Int. Conf. Manament of Data* (June 1996), pp. 205–216.

[16] Imielinski, T., and H.Mannila. A database perspective on knolwedge discovery. *Communications of the ACM 39*, 11 (November 1996), 58–64.

[17] J.Kivinen, and Mannila, H. Approximate dependency inference from relations. *Theoretical Computer Science 149*, 1 (1995), 129–149.

[18] Lin,T.Y.(Program Chair), Ed. *The Fifth International Workshop on Rough Sets and Soft Computing (RSSC'97)* (March 1997).

[19] Mannila, H. Data mining: machine learning, statistics, and databases. In *Proc. of the 8 International Conference on Scientific and Statistical Database Management* (1996), pp. 1–6.

[20] Mannila, H. Methods and problems in data mining. In (to appear) *Proc. of International Conference on Database Theory, Delphi, Greece* (January 1997), F. Afrati and P.Kolaitis, Eds., Springer-Verlag.

[21] Mannila, H., H.Toivonen, and A.Inkeri Verkamo. Efficient algorithms for discoverying association rules. In *Proc. AAAI Workshop on Knowledge Discovery in Databases* (July 1994), pp. 181–192.

[22] Pawlak, Z. Rough sets. *International Journal of Computer and Information Sciences 11*, 5 (1982), 341–356.

[23] Pawlak, Z. *Rough Sets: Theoretical Aspects of Reasoning About Data.* Kluwer Academic Publishers, The Netherlands, 1991.

[24] Pawlak, Z., Grzymala-Busse, J., Slowinski, R., and Ziarko, W. Rough sets. *Communications of the ACM 38*, 11 (1995), 88–95.

[25] Silberschatz, A., and Tuzhilin, A. On subjective measure of interestingness in knowledge discovery. In *Proc. 1st International Conference on Knowledge Discovery and Data Mining (KDD'95) (Montreal, Canada* (Menlo Park, CA., August 1995), AAAI Press, pp. 275–281.

[26] Slowinski, R., and Stefanowski, J. Rough-set reasoning about uncertain data. *Fundamenta Informaticae 27*, 2/3 (1996), 229–243.

[27] Ullman, J. *Principles of Database and Knowledge-base Systems: Volume I: Classical Database Systems.* Computer Science Press, Rockville, MD., 1988.

[28] Ullman, J. *Principles of Database and Knowledge-base Systems: Volume I: Classical Database Systems.* Computer Science Press, Rockville, MD., 1988, pp. 376–446.

[29] Ullman, J. *Principles of Database and Knowledge-base Systems: Volume I: Classical Database Systems.* Computer Science Press, Rockville, MD., 1988, pp. 413–420.

[30] Vardi, M. Fundamentals of dependency theory. In *Trends in Theoretical Computer Science* (Rockville, MD, 1987), E. Borger, Ed., Computer Science Press, pp. 171–224.