A Framework for Measuring Changes in Data Characteristics

Venkatesh Ganti Johannes Gehrke* Raghu Ramakrishnan†
Wei-Yin Loh

Abstract

A data mining algorithm builds a model that captures interesting aspects of the underlying data. We develop a framework for quantifying the difference, called the *deviation*, between two datasets in terms of the models they induce. Our framework covers a wide variety of models including frequent itemsets, decision tree classifiers, and clusters, and captures standard measures of deviation such as the *misclassification rate* and the *chi-squared* metric as special cases. We also show how statistical techniques can be applied to the deviation measure to assess whether the difference between two models is meaningful (i.e., whether the underlying datasets have statistically significant differences in their characteristics), and discuss several practical applications.

1 Introduction

The goal of data mining is to discover (predictive) models based on the data maintained in the database [16]. Several algorithms have been proposed for computing novel models [2, 3, 28, 27, 1], for more efficient model construction [30, 32, 9, 15, 37, 22, 20, 33, 19, 31], and to deal with new data types [20, 18, 23]. There is, however, no work addressing the important issue of how to measure the difference, or *deviation*, between two models.

As a motivating application, a sales analyst who is monitoring a dataset (e.g., weekly sales for Walmart) may want to analyze the data thoroughly only if the current snapshot differs significantly from previously analyzed snapshots. In general, since successive database snapshots overlap considerably, they are quite similar to each other [11, 17, 36]. Therefore, an algorithm that can quantify deviations can save the analyst considerable time and effort.

As a second example, a marketing analyst may want to analyze if and how data characteristics differ across several datasets of customer transactions collected from different stores. The analysis can then be used to decide whether different marketing strategies are needed for each store. Further, based on the deviation between pairs of datasets, a set of stores can be grouped together and earmarked for the same marketing strategy.

In this paper, we develop the `FOCUS` framework for computing an interpretable, qualifiable deviation measure between two datasets to quantify the differences between "interesting" characteristics in each dataset (as reflected in the model it induces when a data mining algorithm is applied on it [16]). The central idea is that a broad class of models can be described in terms of a *structural component* and a *measure component*. The structural component identifies "interesting regions," and the measure component summarizes the subset of the data that is mapped to each region. The `FOCUS` framework has several desirable features:

- The deviation measure obtained from `FOCUS` is intuitively interpretable in terms of the work required to transform one model to the other (Section 3). It can be computed using a single scan of the underlying datasets; a good upper bound for frequent itemsets can be computed by simply examining the models (Section 4.1.1).
- The framework allows comparison of specific parts of two models. This makes it possible to focus attention on interesting changes that might not significantly affect the model as a whole.
- The framework covers the models obtained by several mining algorithms, including frequent itemsets, decision trees, and clusters (Sections 4.1, 4.2, and 4.3). It also captures the *misclassification rate* (commonly used for evaluating decision trees) and *chi-squared statistic* as special cases of the deviation measure. (In Section 5.2.2, we also show how the chi-squared statistic can be applied to decision trees, using the bootstrapping technique to avoid some standard restrictions that would otherwise make it inapplicable.)

We illustrate the power of the framework through these additional contributions:

- We show how `FOCUS` can be used to interactively identify and explore subsets of data that lead to interesting changes in the model being studied (Section 5). We define a set of operators to discover regions where the differences between the datasets are interesting.
- We apply our measure of deviation to study whether models based on a sample of the available data differ

* Supported by an IBM Graduate Fellowship
† This research was supported by Grant 2053 from the IBM corporation.

significantly from the model based on all the data. Interestingly, even for very large sample sizes, there is a statistically significant difference between the sample-based models and the model based on all data. However, the difference diminishes quickly with increasing sample size. In many situations, it may suffice to use a sample, and our measure of deviation can be used to determine the appropriate sample size (Section 6).

2 Preliminaries

We now introduce our notation, beginning with some standard terms. A *partially ordered* set $\langle P; \preceq \rangle$ consists of a non-empty set P and a reflexive, antisymmetric, transitive binary relation $\preceq$ on P. Let $\langle P; \preceq \rangle$ be a partially ordered set and let $H \subseteq P$. An element $a \in P$ is called a *lower bound* of H if $a \preceq h$ for all $h \in H$. A lower bound a of H is the *greatest lower bound* of H if, for any lower bound b of H, we have $b \preceq a$. We denote the greatest lower bound of H by $\bigwedge H$. A partially ordered set $\langle P; \preceq \rangle$ is a *meet-semilattice* if for all $a, b \in P$, $\bigwedge \{a, b\}$ exists.

Let $\mathcal{I} = \{A_1, \ldots, A_n\}$ be a set of attributes. Let $\mathcal{D}_i$ be the *domain* of the attribute A_i, $1 \in \{1, \ldots, n\}$.

Definition 2.1 The *attribute space* $\mathcal{A}(\mathcal{I})$ of $\mathcal{I}$ is the cross product of the domains of all attributes: $\mathcal{D}_1 \times \cdots \times \mathcal{D}_n$. A *region* γ is a subset of the attribute space $\mathcal{A}(\mathcal{I})$; $t = \langle t_1, \ldots, t_n \rangle$ is an n-tuple on $\mathcal{I}$ if $t \in \mathcal{A}(\mathcal{I})$. Each region γ has a corresponding predicate P_γ such that $\{P_\gamma(t)=\mathtt{true}$ iff $t \in \gamma\}$. A *dataset* D is a finite set of n-tuples.

In contrast to a region, a dataset is an enumerated set of tuples in the attribute space. Let $\mathcal{I} = \{A_1, A_2\}$ with domains $[1, 10], [1, 10]$ respectively. $A_1 \leq 5$ and $D = \{\langle 1, 1 \rangle, \langle 2, 1 \rangle\}$ are examples of a region (defined by the predicate) and a dataset respectively.

Definition 2.2 The *selectivity* $\sigma(\gamma, D)$ of a region $\gamma \subseteq \mathcal{A}(\mathcal{I})$ w.r.t. a dataset D is the fraction of tuples in D that fall into γ: $\sigma(\gamma, D) \stackrel{\text{def}}{=} \frac{|\{t : t \in D \wedge t \in \gamma\}|}{|D|}$.

In this paper, we consider three classes of data mining models widely studied in the database literature: lits-models, dt-models, and cluster-models. We use $\mathcal{M}$ generically to denote any one of the three classes. Informally, a lits-model is the set of "frequent" itemsets; a dt-model is a decision tree; a cluster-model is a set of clusters. A formal description of each class of models is given in Appendix A.

3 FOCUS

In this section, we describe the FOCUS framework for computing deviations between the "interesting characteristics" of two datasets. FOCUS can be applied to any class of data mining models that satisfy the *2-Component* and *Meet-semilattice* properties, which are described below. In Section 4, we will show that these properties are satisfied by lits-models, dt-models, and cluster-models.

3.1 2-Component, Meet-semilattice Models

The main idea behind FOCUS is that a model M has a structural component Γ_M that identifies interesting regions of the attribute space, and that each such region is summarized by a measure (e.g., a count). If the structural component satisfies some properties that allow us to "refine" two models naturally, we have the basis for an intuitive and quantitative deviation measure.

Consider an illustrative example. Figures 1, 2, and 3 show example models of a set of frequent itemsets, a decision tree (for 2 classes), and a set of clusters. All three models have one common feature: certain regions in the attribute space are found to be interesting, and are associated with a measure, which is the fraction of tuples in the database that are mapped into the region. Figure 1 shows a simple itemset model where $\mathcal{I} = \{a, b\}$. It has three interesting regions identified by the itemsets $\{a\}$, $\{b\}$, and $\{a, b\}$. Each itemset (equivalently, the corresponding region) is associated with its support: $\{a\}$ with 0.5, $\{b\}$ with 0.4, and $\{a, b\}$ with 0.25. Generalizing from this example, each frequent itemset X in a lits-model represents a region in $\mathcal{A}(\mathcal{I})$ (where the support is higher than the threshold). Figure 2 shows a simple decision tree with 3 leaf nodes and 2 class labels. Each leaf node is associated with two regions, one per class label[1]. Each region is associated with the selectivity of the region. Generalizing from this example, each leaf node of a decision tree for k classes is associated with k regions in $\mathcal{A}(\mathcal{I})$ each of which is associated with its selectivity. Similarly, generalizing from Figure 3, each cluster model has a set of regions each associated with its selectivity.

We now describe the meet-semilattice property, which captures the relationship between models in $\mathcal{M}$. Figure 4 illustrates the relationship between two decision trees T_1 and T_3. The structure of T_3 is "finer" than that of T_1 because we can deduce T_1's measure component with respect to any dataset D if the measure component of T_3 with respect to D is known. Intuitively, T_3 captures information at a finer level than T_1. Similarly, among the two sets of frequent itemsets L_1 and L_3 shown in Figure 5, L_3 is "finer" than L_1 because we can deduce the measure component of L_1 from that of L_3. We capture this relationship between the structural components of two models in $\mathcal{M}$ using a binary relation called the *refinement* relation.

For the classes of models we consider, given two models M_1 and M_2, the greatest lower bound of their structural

[1] The two regions are collapsed into one in Figure 2 instead of adding a third dimension. But their measures are shown separately as an ordered pair.

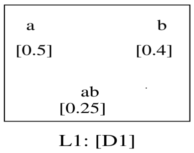

Figure 1: Lits

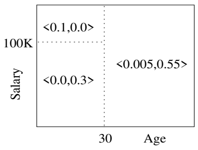

Figure 2: Decision Tree

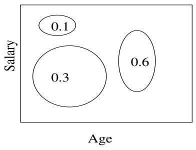

Figure 3: Clusters

components $\Gamma_{M_1}, \Gamma_{M_2}$ under the refinement relation always exists; we call this the *greatest common refinement (GCR)* of Γ_{M_1} and Γ_{M_2}, and denote it by $\Gamma_{\bigwedge(M_1,M_2)}$. The set of all structural components of models in $\mathcal{M}$ along with the refinement relation thus forms a *meet-semilattice*.

Definition 3.1 A class of models $\mathcal{M}$ is said to satisfy the *2-component* property if any $M \in \mathcal{M}$ induced by a dataset D can be described as $\langle \Gamma_M, \Sigma(\Gamma_M, D)\rangle$ where $\Gamma_M = \{\gamma_M^i : 1 \le i \le l\}$ is a set of regions in $\mathcal{A}(\mathcal{I})$ and $\Sigma(\Gamma_M, D) = \{\sigma(\gamma_M^i, D) : \gamma_M^i \in \Gamma_M\}$. We use $\Gamma_{\mathcal{M}}$ to denote the set of structural components of all models in $\mathcal{M}$.

Definition 3.2 Let $\Gamma_{M_1}, \Gamma_{M_2} \in \Gamma_{\mathcal{M}}$. We say that a set of regions $\{\gamma_{j_1}, \ldots, \gamma_{j_k}\}$ *refines* a region γ_i if for any dataset D, $\sigma(\gamma_i, D) = \sum_{i=1}^{k} \sigma(\gamma_{j_i}, D)$. We say that Γ_{M_1} *refines* Γ_{M_2} (denoted $\Gamma_{M_1} \preceq \Gamma_{M_2}$) if for every region $\gamma_{M_2}^j \in \Gamma_{M_2}$ there exists a set of regions $\{\gamma_{M_1}^{j_1}, \ldots, \gamma_{M_1}^{j_{k_j}}\} \subseteq \Gamma_{M_1}$ which refine $\gamma_{M_2}^j$. We call $\preceq$ a *refinement relation*.

Observation 3.1 Let $\mathcal{M}$ be any one of the following three classes of models: lits-models, dt-models, cluster-models. Then $\mathcal{M}$ satisfies the 2-component property and there exists a refinement relation $\preceq$ on $\Gamma_{\mathcal{M}}$ such that $\langle \Gamma_{\mathcal{M}}; \preceq\rangle$ is a meet-semilattice.

This observation summarizes results in Sections 4.1 and 4.2.

3.2 Measuring Deviations

We now develop our measure of deviation between two models M_1 and M_2, and thereby, between the underlying two datasets. Intuitively, the difference between the models is quantified as the amount of work required to transform one model into the other, which is small if the two models are "similar" to each other, and high if they are "different."

When the structural components are identical we can transform the measure component of one model to the other by making the measure at each region under the first model agree with that under the second model. Let $\Gamma_{M_1} = \Gamma_{M_2}$. Then, the amount of work for transforming $\Sigma(\Gamma_{M_1}, D_1)$ into $\Sigma(\Gamma_{M_2}, D_2)$ is the aggregate of the differences between $\sigma(\gamma_{M_1}^i, D_1)$ and $\sigma(\gamma_{M_2}^i, D_2)$, $i = 1, \ldots, |\Gamma_{M_1}|$. We assume that the difference, at a region, between the measures of the first and the second models is given by a *difference function* f (not necessarily the usual difference operator "-"), and that the aggregate of the differences is given by an *aggregate function* g. We discuss these functions, which enhance FOCUS's ability to instantiate deviation functions for specialized applications, in Section 3.2.2. For now, it suffices to say that f and g are model-independent parameters of FOCUS with the signatures $f : \mathcal{I}_+^4 \mapsto \mathcal{R}_+$, and $g : \mathcal{P}(\mathcal{R}_+) \mapsto \mathcal{R}_+$[2].

We now formally define the deviation when the structural components of the two models are identical.

Definition 3.3 Let f be a difference function, g an aggregate function, and $M_1, M_2 \in \mathcal{M}$ be two models induced by the datasets D_1, D_2 respectively, such that $\Gamma_{M_1} = \Gamma_{M_2} = \{\gamma_1, \ldots, \gamma_l\}$. For $j \in \{1,2\}$, let $\kappa_{D_j}^i = \sigma(\gamma_i, D_j) \cdot |D_j|$ denote the absolute number of tuples in D_j that are mapped into $\gamma_{M_j}^i \in \Gamma_{M_j}$. The deviation between M_1 and M_2 is defined as follows:

$$\delta^1_{(f,g)}(M_1, M_2) \stackrel{\text{def}}{=} g(\{f(\kappa^1_{D_1}, \kappa^1_{D_2}, |D_1|, |D_2|), \ldots, f(\kappa^l_{D_1}, \kappa^l_{D_2}, |D_1|, |D_2|)\})$$

In general, two models induced from different datasets have significantly different structural components. Therefore we first have to reconcile the differences in the structural components of two models to make them comparable. To do this, we rely on the meet-semilattice property exhibited by many classes of data mining models (see Observation 3.1). The idea is to "extend" both models to the GCR of their structural components, and then compare the extensions. Intuitively, to extend a model M to $\Gamma_{M'}$ ($\preceq \Gamma_M$) we find the measure component $\Sigma(\Gamma_{M'}, D)$ for $\Gamma_{M'}$ using the dataset D, i.e., we find the selectivity of each region in $\Gamma_{M'}$ w.r.t. D.

Definition 3.4 Let $M_1, M_2 \in \mathcal{M}$ be two models induced by D_1, D_2 respectively. We define the deviation $\delta_{(f,g)}(M_1, M_2)$ between M_1 and M_2 as follows: $\delta_{(f,g)}(M_1, M_2) \stackrel{\text{def}}{=}$

$$\delta^1_{(f,g)}(\langle \Gamma_{\bigwedge(M_1,M_2)}, \Sigma(\Gamma_{\bigwedge(M_1,M_2)}, D_1)\rangle, \langle \Gamma_{\bigwedge(M_1,M_2)}, \Sigma(\Gamma_{\bigwedge(M_1,M_2)}, D_2)\rangle)$$

Usually, we drop f and g because they are clear from the context.

For certain choices of f and g (identified in Sections 4.1 and 4.2), using the GCR gives the least value for δ over all common refinements. This property of the least deviation then corresponds to the least-work transformation between the two models.

Summarizing, the instantiation of FOCUS requires:

1. A refinement relation $\preceq$.
2. A difference function f and an aggregate function g.

[2] $\mathcal{I}_+$ and $\mathcal{R}_+$ denote the sets of non-negative integers and non-negative real numbers respectively.

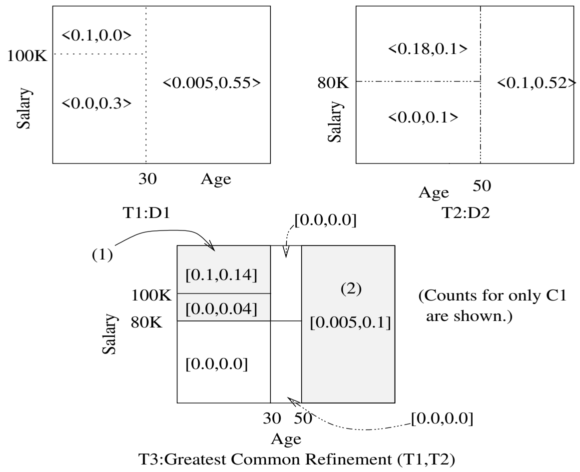

Figure 4: dt-model:$T_3 = \bigwedge(T_1, T_2)$

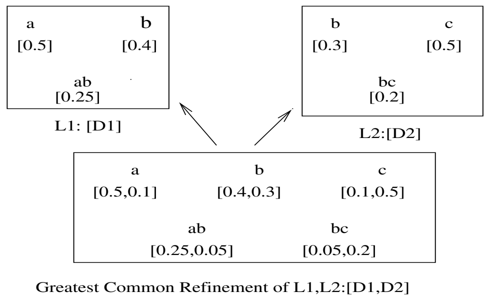

Figure 5: lits-model: $L_3 = \bigwedge(L_1, L_2)$

3.2.1 Computational Requirements for δ

The computation of $\delta(M_1, M_2)$ requires the selectivities of all regions in $\Gamma_{\bigwedge(M_1,M_2)}$ to be computed w.r.t. both the datasets D_1 and D_2. For the three classes of data mining models we consider, this requires D_1 and D_2 to be scanned once.

3.2.2 Difference and Aggregate Functions

In this section, we motivate the use of parameters f and g in the FOCUS framework. We then present two example instantiations each for f and g.

We first consider f. Let L_1 and L_2 be two lits-models induced by D_1 and D_2. Without loss of generality, let us assume that L_1 and L_2 have identical structural components Γ. (Otherwise, we can extend them to their GCR.) Consider two itemsets X_1 and X_2 in Γ. Suppose $\sigma(\gamma_{L_1}^{X_1}, D_1) = 0.5, \sigma(\gamma_{L_2}^{X_1}, D_2) = 0.55$, and $\sigma(\gamma_{L_1}^{X_2}, D_1) = 0.0, \sigma(\gamma_{L_2}^{X_2}, D_2) = 0.05$. So, X_1 varies between a "significant" 50% and a "more significant" 55% whereas X_2 varies between a "non-existent" 0% and a "noticeable" 5%. For some applications, the variation in X_2 is more significant than that in X_1 because noticing an itemset for the first time is more important than a slight increase in an already significant itemset. For some other applications which just concentrate on the absolute changes in support, the variations in X_1 and X_2 are equally important. To allow both cases, our first instantiation f_a finds the absolute difference between the supports, while the second instantiation f_s "scales". We now define the two instantiations[3].

Definition 3.5 Let $\kappa_1, \kappa_2, N_1, N_2 \in \mathcal{I}_+$ such that $\kappa_1 < N_1$ and $\kappa_2 < N_2$. The *absolute difference function* and the *scaled difference function* are defined as follows:

$$f_a(\kappa_1, \kappa_2, N_1, N_2) \stackrel{\text{def}}{=} \left|\frac{\kappa_1}{N_1} - \frac{\kappa_2}{N_2}\right|$$

$$f_s(\kappa_1, \kappa_2, N_1, N_2) \stackrel{\text{def}}{=} \begin{cases} \frac{|\frac{\kappa_1}{N_1} - \frac{\kappa_2}{N_2}|}{(\frac{\kappa_1}{N_1} + \frac{\kappa_2}{N_2})/2}, & \text{if } (\kappa_1 + \kappa_2) > 0 \\ 0, & \text{otherwise} \end{cases}$$

The aggregate function g takes as input a set of values. The two most commonly used aggregate functions are *sum* and *max*. Since the instantiations of f and g are independent of each other, these example instantiations generate four different instantiations of δ.

3.3 The Qualification Procedure

Suppose the deviation between D_1 and D_2 is 0.005, and that between D_1 and D_3 is 0.01. From just the deviation

[3] The signature $f : \mathcal{R}_+ \times \mathcal{R}_+ \mapsto \mathcal{R}_+$ for f where the two arguments correspond to the selectivities of a region w.r.t. both datasets suffices for most purposes. However, some functions require absolute measures. We give one such example in Section 5.2.2. Therefore, we use absolute measures.

values, we are able to say the data characteristics of D_1 and D_2 are more similar than those of D_1 and D_3. But, we still do not know whether they have "different" data characteristics; a deviation of 0.01 may not be uncommon between two datasets generated by the same process. In other words, is the deviation value statistically "significant"?

The availability of a quantitative deviation measure makes it possible to answer such questions rigorously. If we assume that the distribution $\mathcal{F}$ of deviation values under the hypothesis that the two datasets are generated by the same process is known, we can use standard statistical tests to compute the *significance* $sig(d)$ of the deviation d between two datasets. We use *bootstrapping* techniques from Statistics [14] to compute $\mathcal{F}$. For completeness, we present the details of the statistical tests and the bootstrapping procedure in Appendix C.

4 Instantiations

In this section, we instantiate the `FOCUS` framework for lits-models, dt-models, and cluster-models. Wherever possible, we analyze the properties of the instantiated deviation functions.

4.1 lits-models

We first show that the class of lits-models exhibits the meet-semilattice property. Next, we analyze the deviation functions and discuss interesting characteristics that arise due to the use of the GCR. We then derive an upper bound for the deviation functions $\delta_{(f_a,g)}$ where $g \in \{g_{sum}, g_{max}\}$.

The refinement relation between the structural components of two sets of frequent itemsets is defined by the *superset* relation. Let $\Gamma_{M_1} = L_{D_1}^{m_s}$ and $\Gamma_{M_2} = L_{D_2}^{m_s}$ be two sets of frequent itemsets[4]. Formally, $\Gamma_{M_1} \preceq_L \Gamma_{M_2}$ if $L_{D_1}^{m_s} \supseteq L_{D_2}^{m_s}$. The powerset of a set of objects (here, $\mathcal{I}$) along with the superset relation forms a meet-semilattice [21]. (In fact, it forms a lattice.)

Proposition 4.1 The class of lits-models $\mathcal{M}$ on the set of items $\mathcal{I}$ exhibits the 2-component property and $\langle \Gamma_{\mathcal{M}}; \preceq_L \rangle$ is a meet-semilattice.

Consider the example in Figure 5. L_3 is the greatest common refinement of L_1 and L_2. The supports from D_1 and D_2 for each itemset in the greatest common refinement are shown below it. $\delta_{(f_a,g_{sum})}(L_1, L_2) = 0.4 + 0.1 + 0.4 + 0.2 + 0.15 = 1.125$, and $\delta_{(f_a,g_{max})}(L_1, L_2) = 0.4$.

We now show that using the GCR of two models rather than any common refinement gives the least deviation.

Theorem 4.1 Let $f \in \{f_a, f_s\}$ and $g \in \{g_{sum}, g_{max}\}$. Let Γ_M be a common refinement of Γ_{M_1} and Γ_{M_2}. Then,

$$\delta(M_1, M_2) \leq \delta^1_{(f,g)}(\langle \Gamma_M, \Sigma(\Gamma_M, D_1)\rangle, \langle \Gamma_M, \Sigma(\Gamma_M, D_2)\rangle)$$

[4] $L_{D_1}^{m_s}$ is the set of itemsets in D_1 with support greater than m_s. See Appendix A for a definition of lits-models.

4.1.1 Upper Bound δ^* for δ

In an exploratory, interactive environment where δ is repeatedly computed, we can typically work with just estimates of the actual answers, but require fast responses. For the case where the difference function is f_a, we now derive an upper bound δ^* of δ that can be computed fast using just the two models (which will probably fit in main memory, unlike the datasets). Using the upper bound δ^* instead of δ is safe; we will not ignore significant deviations. δ^* also satisfies the *triangular inequality*, and can therefore be used to embed a collection of datasets in a k-dimensional space for visually comparing their relative differences.

Definition 4.1 Let $\mathcal{M}$ be the class of lits-models and $M_1, M_2 \in \mathcal{M}$ be two models at minimum support level m_s induced by D_1 and D_2. Let $\kappa_1, \kappa_2 \in \mathcal{I}_+$. Let $f^*(\kappa_1, \kappa_2, |D_1|, |D_2|)$

$$\stackrel{\text{def}}{=} \begin{cases} f_a(\kappa_1, \kappa_2, |D_1|, |D_2|), \text{ if } \frac{\kappa_1}{|D_1|}, \frac{\kappa_2}{|D_2|} > m_s \\ f_a(\kappa_1, 0, |D_1|, |D_2|), \text{ if } \frac{\kappa_1}{|D_1|} > m_s \text{ and } \frac{\kappa_2}{|D_2|} < m_s \\ f_a(0, \kappa_2, |D_1|, |D_2|), \text{ if } \frac{\kappa_1}{|D_1|} < m_s \text{ and } \frac{\kappa_2}{|D_2|} > m_s \end{cases}$$

We define $\delta^*_{(g)}(M_1, M_2) \stackrel{\text{def}}{=} \delta_{(f^*,g)}(M_1, M_2)$.

Theorem 4.2 Let $M_1, M_2 \in \mathcal{M}$ be two models induced by D_1, D_2 and let $g \in \{g_{sum}, g_{max}\}$. Then the following properties hold:

(1) $\delta^*_{(g)}(M_1, M_2) \geq \delta_{(f_a,g)}(M_1, M_2)$
(2) $\delta^*_{(g)}$ satisfies the triangular inequality.
(3) $\delta^*_{(g)}$ can be computed without scanning D_1 or D_2.

4.2 dt-models

For the rest of the section, let $M_1, M_2 \in \mathcal{M}$ be two dt-models induced by D_1, D_2 respectively, and P_γ denote the predicate identifying a region γ.

Definition 4.2 $\Gamma_{M_1} \preceq_T \Gamma_{M_2}$ if, $\forall \gamma^i_{M_2} \in \Gamma_{M_2}, \exists \{\gamma^{i_1}_{M_1}, \ldots, \gamma^{i_{j_i}}_{M_1}\} \subseteq \Gamma_{M_1}$: $\{(P_{\gamma^{i_1}_{M_1}} \vee \cdots \vee P_{\gamma^{i_{j_i}}_{M}}) \text{ iff } P_{\gamma^i_{M_2}}\}$.

Intuitively, the GCR of the structural components of two dt-models is the finer partition of $\mathcal{A}(\mathcal{I})$ obtained by overlaying the two structural components Γ_{M_1} and Γ_{M_2}. The corresponding set of predicates is obtained by "anding" all possible pairs of predicates from both the structural components. For example, Figure 4 illustrates the finer partition formed by overlaying the partitions of the models T_1 and T_2. For the sake of clarity, we show the measures only for regions of class label C_1 in the GCR. (An identical structure exists for the second class label.) Formally, the GCR $\Gamma_{\bigwedge(M_1,M_2)}$ of Γ_{M_1} and Γ_{M_2} is:

$$\{\gamma : \gamma \text{ is identified by } P_{\gamma_1} \wedge P_{\gamma_2} \ni \gamma_1 \in \Gamma_{M_1} \wedge \gamma_2 \in \Gamma_{M_2}\}$$

Proposition 4.2 Let $\mathcal{M}$ be the class of dt-models with refinement relation $\preceq_T$. Then $\mathcal{M}$ exhibits the 2-component property and $\langle \Gamma_{\mathcal{M}}; \preceq_T \rangle$ is a meet-semilattice.

Consider the example in Figure 4. T_3's structural component is the GCR of the structural components of T_1 and T_2. For the sake of clarity, only the measures of class C_1 from both D_1 and D_2 are shown in T_3. $\delta_{(f_a, g_{sum})}(T_1, T_2)$ over regions corresponding to class C_1 is: $|0.1 - 0.14| + |0.0 - 0.04| + |0 - 0| + |0 - 0| + |0 - 0| + |0.005 - 0.1| = 0.175$.

The following theorem shows that using the greatest common refinement, rather than any common refinement, gives the least deviation value for the case $g = g_{sum}$.

Theorem 4.3 Let Γ_M be a common refinement of Γ_{M_1} and Γ_{M_2}. Let $g = g_{sum}$, and $f \in \{f_a, f_s\}$. Then, $\delta_{(f, g_{sum})}(M_1, M_2) \leq$
$\delta^1_{(f,g)}(\langle \Gamma_M, \Sigma(\Gamma_M, D_1)\rangle, \langle \Gamma_M, \Sigma(\Gamma_M, D_2)\rangle)$

Observe that this theorem is less general than Theorem 4.1 for lits-models. We give a counter example in Appendix B to illustrate that the above lemma is not valid for $g = g_{max}$.

4.3 cluster-models

The instantiation for the class of cluster-models is just a special case of the instantiation of FOCUS for dt-models. We omit the details due to space constraints.

5 Focussed Deviations

In this section, we illustrate the power of the FOCUS framework by applying it to two different scenarios: *exploratory analysis* and *change monitoring*. The objective in the first setting is to interactively explore and understand the differences between two datasets, similar to the drill-down and roll-up strategies in OLAP databases [12] and the *ad hoc mining* approach emphasized in [29, 25]. The objective in the second setting is to check how well a model built from an old dataset fits a new dataset.

For both application scenarios, a very useful property of FOCUS is that we can compute deviations w.r.t. a specific region $\gamma \subseteq \mathcal{A}(\mathcal{I})$. Each region in the structural component $\Gamma_M = \{\gamma^i_M, i = 1, \ldots, |\Gamma_M|\}$ of the model M can be independently focussed w.r.t. γ by taking its intersection with γ. The measure w.r.t. a dataset D for each region γ^i_M focussed w.r.t. γ is $\sigma(\gamma \cap \gamma^i_M, D)$.

Definition 5.1 Let $M \in \mathcal{M}$ be a model induced by the dataset D and $\gamma \subset \mathcal{A}(\mathcal{I})$ be a region, called the *focussing region*. Then the *focus* of M w.r.t. γ is defined as:

$$M^\gamma \stackrel{\text{def}}{=} \langle \Gamma^\gamma_M, \Sigma(\Gamma^\gamma_M, D)\rangle$$

where $\Gamma^\gamma_M = \{\gamma \cap \gamma^i_M : \gamma^i_M \in \Gamma_M\}$. We use $\mathcal{M}^\gamma$ and $\Gamma^\gamma_{\mathcal{M}}$ to denote the sets of all models in $\mathcal{M}$ and structural components in $\Gamma_{\mathcal{M}}$ focussed w.r.t. γ.

The following theorem shows that all the theory developed for the class of models $\mathcal{M}$ can be applied to $\mathcal{M}^\gamma$ as well.

Theorem 5.1 Let $\mathcal{M}$ be one of the following three classes of models: lits-models, dt-models, and cluster-models. Let $\preceq$ be a refinement relation such that $\langle \Gamma_{\mathcal{M}}; \preceq \rangle$ forms a meet-semilattice. Let $\gamma \subseteq \mathcal{A}(\mathcal{I})$ be the focussing region. Then $\langle \Gamma^\gamma_{\mathcal{M}}; \preceq \rangle$ is a meet-semilattice.

Definition 5.2 Let f be a difference function, g an aggregate function, and M_1, M_2 be two models induced by D_1, D_2, respectively. The deviation $\delta^\gamma_{(f,g)}(M_1, M_2)$ between M_1 and M_2 focussed w.r.t. a region $\gamma \subseteq \mathcal{A}(\mathcal{I})$ is defined as:

$$\delta^\gamma_{(f,g)}(M_1, M_2) \stackrel{\text{def}}{=} \delta_{(f,g)}({M_1}^\gamma, {M_2}^\gamma)$$

We emphasize that the deviation function may not be monotonic, i.e., if $\gamma \subset \gamma'$ then the deviation over γ may not be less than that over γ'. For example, if M_1, M_2 are two models constructed from D_1, D_2 respectively and $g \in \{g_{sum}, g_{max}\}$ then $\gamma \subseteq \gamma' \Rightarrow \delta^\gamma_{(f_a,g)}(M_1, M_2) \leq \delta^{\gamma'}_{(f_a,g)}(M_1, M_2)$. However, the same is not true for $\delta_{(f_s,g)}(M_1, M_2)$.

The ability to compute region-specific deviations is enhanced by adding operators to manipulate sets of regions. We now introduce a small collection of such operators, divided into two groups: *structural* and *rank* operators.

1. `Structural Union (⊔):` The structural union of two sets of regions Γ_1 and Γ_2 is given by their GCR $\bigwedge(\Gamma_1, \Gamma_2)$.
2. `Structural Intersection (⊓):` The structural intersection of Γ_1 and Γ_2 is the set of regions Γ such that each region in Γ is a member of both Γ_1 and Γ_2. This is identical to the standard intersection operation on sets.
3. `Structural Difference ⊖:` The structural difference of Γ_1 and Γ_2 is $(\Gamma_1 \sqcup \Gamma_2) - (\Gamma_1 \sqcap \Gamma_2)$.
4. `Predicate` p: The predicate region is a subset of the attribute space identified by p.

Given a set of regions, the rank operator orders them by the "interestingness" of change between the two datasets. The interestingness of a region is captured by a deviation function.

- `Rank`: Given a set of regions Γ, two datasets D_1, D_2, and a deviation function $\delta_{(f,g)}$, the `rank` operator $\rho(\Gamma, \delta_{(f,g)}, D_1, D_2)$[5] returns as output a list $\vec{\Gamma}$ of regions in the decreasing order of interestingness.
- `Select`: Given the output of the rank operator, the selection operator selects a subset of the output. For example, `top-region`, `top-n regions`, `min-region`, and `bottom-n regions` are common selections; we denote these selections by θ^{top}, θ^{n}, θ^{min}, and θ^{-n} respectively.

5.1 Exploratory Analysis

The objective in exploratory analysis is to find a set of interesting regions in terms of the differences between the two datasets. Consider the decision trees T_1 and T_2 constructed from D_1 and D_2 shown in Figure 4. Suppose that deviations above `0.05` are considered significant. D_1 and D_2 differ considerably in the shaded regions `(1)` and `(2)`. If $f = f_a$ then these regions have a deviation (w.r.t. class C_1) of `0.08` and `0.095` respectively. Note that region `(1)` is a leaf node of T_1 but region `(2)` is a sub-region of a leaf node in T_2. Moreover, the sub-regions of `(1)` in T_3 do not cause significant differences between D_1 and D_2. Therefore, we have to find regions that are significantly different at all levels of the tree in addition to the regions of T_3. The following expressions find the regions `(1)` and `(2)` respectively:

$$\theta^{top}(\rho(\Gamma_{T_1} \sqcup \Gamma_{T_2}, \delta_{(f_a, g_{sum})})), \quad \theta^{top}(\rho(\Gamma_{T_1} \sqcup \Gamma_{T_2}, \delta_{(f_a, g_{sum})}))$$

Next, consider an example in the frequent itemset domain. The shoes and clothes departments in the Walmart super market sell sets of items $\mathcal{I}_1$ and $\mathcal{I}_2$ respectively. Suppose D_1 and D_2 are datasets collected at two different outlets. An analyst compares the top-10 itemsets in each department to see if the popular itemsets are similar across the two departments. Let L_1 and L_2 be the sets of frequent itemsets computed from D_1 and D_2 respectively. Let f and g be chosen appropriately. The following expressions return the top-10 lists from each department, and the combined top-20:

$$\rho(\theta^{10}(\rho(\mathcal{P}(\mathcal{I}_1) \cap (\Gamma_{L_1} \sqcup \Gamma_{L_2})), \delta) \cup \theta^{10}(\rho(\mathcal{P}(\mathcal{I}_2) \cap (\Gamma_{L_1} \sqcup \Gamma_{L_2}))), \delta)$$

$$\theta^{20}(\rho(\mathcal{P}(\mathcal{I}_1) \cup \mathcal{P}(\mathcal{I}_2)) \cap (\Gamma_{L_1} \sqcup \Gamma_{L_2}), \delta)$$

5.2 Monitoring Change

The objective in this setting is to know how well the model constructed from the old dataset fits the new dataset. Therefore, the structural component for the model on the new dataset is expected to be that of the old dataset, and the question can be cast as "By how much does the old model misrepresent the new data?" For decision trees, the misclassification error is widely used for this purpose; as we show, the chi-squared metric can also be adapted (using bootstrapping) to address this question. We show that these two traditional measures can be captured as special cases of the `FOCUS` framework by appropriate choices of f and g. Thus, `FOCUS` generalizes change monitoring in two ways: (1) to models other than decision trees, and (2) to change monitoring over specific regions.

[5] Since D_1 and D_2 are usually clear from the context, we omit them from the notation.

5.2.1 Misclassification Error

Let $T = \langle \Gamma_T, \Sigma(\Gamma_T, D_1) \rangle$ be a dt-model constructed on the dataset D_1, and let D_2 be an independent dataset. For each tuple $t \in D_2$, let $C' = T(t)$ be the class label predicted by T for t. If the true class C of t is different from C' then t is said to be *misclassified* by T. The misclassification error $ME^T(D_2)$ of T w.r.t. D_2 is the fraction of the number of tuples in D_2 misclassified by T.

$$ME^T(D_2) \stackrel{\text{def}}{=} \frac{|\{t \in D_2 \text{ and } T \text{ misclassifies } t\}|}{|D_2|}$$

We define the *predicted dataset* D_2^T of D_2 w.r.t. T to be the set of tuples formed by replacing the class label of each tuple $t \in D_2$ with T's prediction for t. Denoting the replacement of the class label of a tuple t with c by $t|c$,

$$D_2^T \stackrel{\text{def}}{=} \{t' : t' = t|T(t), t \in D_2\}$$

The following theorem shows that $ME^T(D_2)$ is the deviation between D_2 and D_2^T at Γ_T.

Theorem 5.2 Let T be a dt-model induced by D_1. Let D_2 be another dataset. Then $ME^T(D_2)$=

$$\frac{1}{2}\delta_{(f_a, g_{sum})}(\langle \Gamma_T, \Sigma(\Gamma_T, D_2) \rangle, \langle \Gamma_T, \Sigma(\Gamma_T, D_2^T) \rangle)$$

5.2.2 Chi-squared Goodness of Fit Statistic

The computation of the chi-squared statistic X^2 assumes that the entire space is partitioned into cells each of which is associated with "expected" and "observed" measures. (See Appendix D for details.) To apply the chi-squared test to dt-models, we use the regions associated with a decision tree T as the cells since these regions partition the entire attribute space. The expected and observed measures are: $E(\gamma_i, D_2) = \sigma(\gamma_i, D_1) \cdot |D_2|, O(\gamma_i, D_2) = \sigma(\gamma_i, D_2) \cdot |D_2|$. The statistic X^2 can now be computed in a straightforward way except for two problems:

(1) For the chi-squared statistic to be well-defined, $E(\gamma_i, D_2)$ should not be zero. We follow the standard practice in Statistics and add a small constant $c > 0$ (0.5 is a common choice) to ensure this [13].

(2) At least 80% of the expected counts must be greater than 5 in order to use the standard X^2 tables (which assume that X^2 values follow a normal approximation to the

binomial distribution). In a decision tree, this condition is often violated. For example, if all tuples in node n are of class i, the expected measures for regions $\gamma_j^n, j \neq i$ will be zero. The solution to this problem is to use an exact calculation for the probability distribution of the X^2 statistic under the null hypothesis, i.e., distribution of X^2 values when the new dataset fits the old model [13]. The procedure (see Section 3.3) to estimate the exact distribution using the bootstrapping technique can be used to perform the test.

It is easy to show that chi-squared statistic, adapted as described above, can be instantiated from FOCUS.

Proposition 5.1 Let T be the decision tree induced by D_1, and let D_2 be another dataset. Let c be a (small) constant. Then the chi-squared statistic X^2 is given by:

$$X^2 = \delta_{(f,g_{sum})}(\langle T, \Sigma(T, D_1)\rangle, \langle T, \Sigma(T, D_2)\rangle) \text{ where}$$

$$f(v_1, v_2, |D_1|, |D_2|) = \begin{cases} \frac{|D_2|(\frac{v_1}{|D_1|} - \frac{v_2}{|D_2|})^2}{\frac{v_1}{|D_1|}}, \text{ if } v_1 > 0 \\ c, \text{ otherwise} \end{cases}$$

6 Effect of Sample Size

In this section, we address the following question. *While constructing a model using a random sample of the dataset, do bigger sample sizes necessarily yield better models?* We apply FOCUS to quantify the notion of "representativeness" of a random sample in inducing the "true" model induced by the entire dataset.

The intuition behind our approach is as follows. The deviation obtained from an instantiation of FOCUS quantifies the difference between the models induced by two datasets. If one of the datasets is a sample randomly drawn from the other, the deviation between the models they induce is then a measure of the *representativeness* of the sample in inducing the true model.

Let M be the model induced by D, and M_S the model induced by a random sample S drawn from D. We define the *sample deviation (SD)* of S to be $\delta(M, M_S)$. The smaller the SD of S, the more representative S is of D. This definition gives us a handle to study the influence of the size of the sample on its representativeness.

Using the SD, we now address two questions. First, does increasing the size of the sample decrease its SD? If so, by how much? If the answer to the first question is affirmative, then the SDs of two sample sizes can be compared to answer the second question; in Sections 6.1.1 and 6.1.2, we carry out this comparison for a wide variety of datasets and models. If the answer to the first question is negative, then the second question is irrelevant. We now describe a procedure that returns the statistical *significance* of the decrease in SD due to an increase in the sample size. The significance is the percentage confidence $100(1-\alpha)\%$ with which the null hypothesis that the two sample sizes are equally representative is rejected.

The basic intuition behind the procedure is as follows. Consider two sets of random samples where the first set S_1 contains samples of size s_{i+1}, and the second set S_2 contains samples of size s_i $(< s_{i+1})$. If the SD measures for size s_{i+1} is smaller than that of s_i $(< s_{i+1})$ then we expect a large number of SD values for S_1 to be smaller than those for S_2. We use the Wilcoxon two-sample test to check the significance of this hypothesis [7]. For the sake of completeness, we describe this procedure in Appendix E.

6.1 Empirical Study

In this section, we present an empirical study of the representativeness of a sample versus its size for lits-models and dt-models.

6.1.1 lits-models

We used the synthetic data generator from the IBM Quest Data Mining group[6]. We use NM.t_lL.$|\mathcal{I}|$I.$N_p$pats.$p$plen to refer to a dataset with N transactions (in multiples of millions), average transaction length t_l, $|\mathcal{I}|$ items (in multiples of 1000's), N_p patterns (in multiples of 1000's), and average pattern length p. We used the Apriori algorithm [5] to compute the set of frequent itemsets from a dataset.

We studied all four combinations of f and g. Due to space constraints, we only present the results of $\delta_{(f_a,g_{sum})}$. The remaining plots are given in Appendix F. We varied two parameters: the size of the dataset and the minimum support level. The datasets used for this study have three different sizes: 1 million, 0.75 million, and 0.5 million transactions. All other parameters to the data generator are set as follows: $|\mathcal{I}| = 1000, t_l = 20, N_p = 4000, p = 4$. Figures 6, 7, and 8 show the sample deviation (SD) versus the sample fraction (SF) values. We draw the following conclusions. (1) As the minimum support level decreases, the size of the sample required to achieve a certain level of representativeness increases. This is to be expected because the lower the minimum support level the more difficult it is to estimate the model. (2) For a given SF value, the representativeness of samples of a fixed size increases with the dataset size. Again, this is as expected.

Table 1 shows the significance of the decrease in SD for the dataset 1M.20L.1I.4pats.4plen as we increase the sample size. We measured the significance using the Wilcoxon test on sets of 50 sample deviation values for each size. We conclude that the representativeness of samples increases with the size of the sample. However, from Figures 6, 7, and 8 we see that the decrease in SD is not high when the sizes of the sample relative to the dataset size (SF) are larger than 30%.

[6] Available from http://www.almaden.ibm.com/cs/quest/syndata.html.

Sample Fraction	0.01	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Significance	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	99.99	-

Table 1: lits-models:% significance of increase in representativeness with sample size from s_i to s_{i+1}

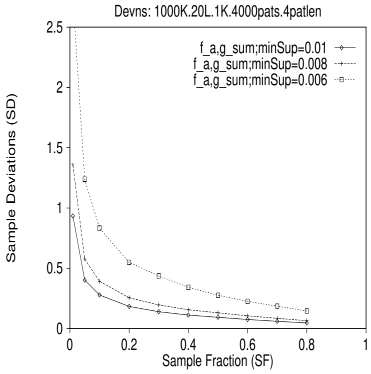

Figure 6: SD vs SF

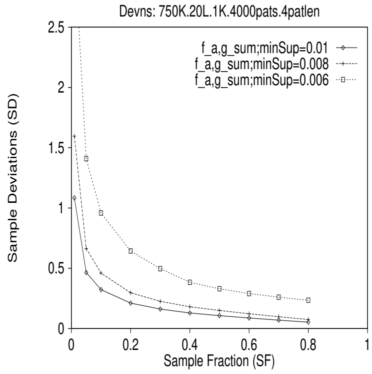

Figure 7: SD vs SF

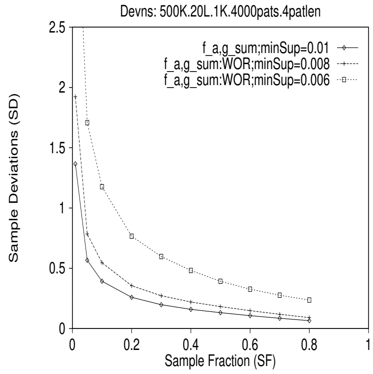

Figure 8: SD vs SF

6.1.2 dt-models

We use the synthetic generator introduced in [2]. It has several classification functions to generate datasets with different characteristics. We selected four functions (Functions F1, F2, F3, and F4) for our performance study. We use NM.Fnum to denote a dataset with N million tuples generated using classification function num. We used a scalable version of the widely studied CART [8] algorithm implemented in the RainForest framework [19] to construct decision tree models. We used $\delta_{(f_a,g_{sum})}$ to compute the deviation between two models.

Table 2 shows the significance of the decrease in sample deviations for dataset 1M.F1 as the sample size is increased. The significance is measured using the Wilcoxon test on sets of 50 sample deviation values for each sample size. The decrease in sample deviation values is quite significant even at SF=70%.

Figures 9, 10, and 11 show the plots for different classification functions (F1, F2, F3, and F4) in the IBM data generator and for different dataset sizes.

6.1.3 Conclusions from this study

For both classes of models, based on the significance values from the Wilcoxon tests, we conclude that it is better to use larger samples because the decrease in sample deviations is statistically significant even for sample sizes as large as 70-80%. On the other hand, the SD versus SF plots suggest that the rate of additional information obtained decreases with increasing sample size, and for many applications, it may be sufficient to take a sample of size 20-30% of the original dataset.

7 Experimental Evaluation

In this section, we evaluate the deviation computation and significance detection algorithms in two parts: first for lits-models and then for dt-models. The datasets we used for this study are also generated from the IBM data generators described in Section 6.1, and the naming conventions are the same as in Section 6.1.

7.1 Set of Frequent Itemsets

In this section, through controlled experiments on synthetic datasets, we first evaluate the procedure for detecting significant deviations. We then evaluate the quality and speed of the upper bound of the deviation function δ^*.

Let D=1M.20L.1I.4pats.4plen. We compute deviations between D and a variety of datasets. All datasets $D_{(1)} - D_{(7)}$ are generated with an average transaction length 20, and 1000 items; $D_{(1)}$ consists of 500K transactions, $D_{(2)} - D_{(4)}$ consist of a million transactions each, and $\delta_{(5)} - \delta_{(7)}$ consist of 50K transactions each. The number of patterns and the average pattern length for each dataset is as follows. $D_{(1)}$: (4K,4); $D_{(2)}, \delta_{(5)}$: (6K,4); $D_{(3)}, \delta_{(6)}$: (4K,5); $D_{(4)}, \delta_{(7)}$: (5K,5). In each case, we set the minimum support level to 1% to compute the set of frequent itemsets from both datasets. Figure 12 shows the deviation values and their significance. The deviation value $\delta_{(f_a,g_{sum})}$ and its significance in row (1) reflect the fact that $D_{(1)}$ has the same distribution as that of D. As expected, $D_{(2)}, D_{(3)}, D_{(4)}$ differ significantly from D. Moreover, the deviation values suggest that the parameter patlen has a large influence on data characteristics. The addition of $\delta_{(5)}$ and $\delta_{(6)}$ to D (rows (6),(7)) cause significant deviations because they differ in the patlen parameter whereas the addition of $\delta_{(7)}$ which differs only in the parameter pats does not cause a significant deviation

Sample Fraction	0.01	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Significance	99.99	99.99	99.99	99.97	99.69	79	99.22	99.93	95.25	-

Table 2: dt-models:% significance of decrease in sample deviation with sample fraction from s_i to s_{i+1}

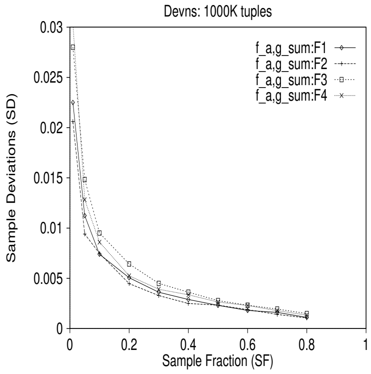

Figure 9: SD vs SF

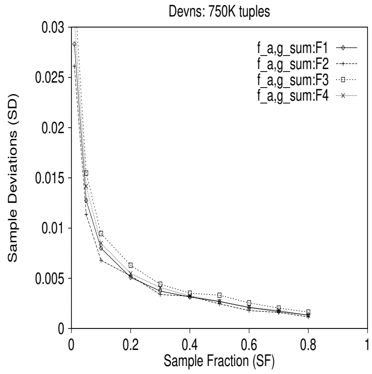

Figure 10: SD vs SF

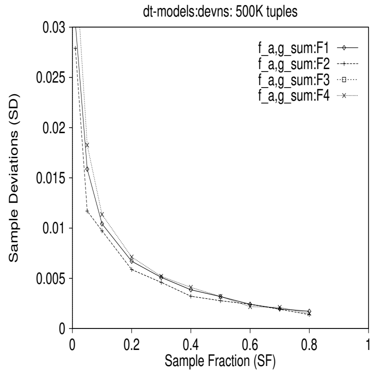

Figure 11: SD vs SF

(row (5)).

The last three columns in Figure 12 show that δ^* delivers a good estimate instantaneously. The lack of difference between the times in the row (1) is due to the fact that D and $D_{(1)}$ have identical distributions. Therefore, the sets of frequent itemsets were identical; so all the measures necessary to compute the deviation are obtained directly from the models.

7.2 Decision Tree Classifiers

We evaluate the significance detection procedure (see Section 3.3) for dt-models using the same experimental framework as in Section 6.1.2. In this experiment, we compute the deviations using $\delta_{(f_a, g_{sum})}$ and their significance values between D=1M.F1 and a variety of datasets. The datasets for the first four rows are generated using the functions F1, F2, F3, and F4 respectively. The datasets used for the last three rows are obtained by extending D with a new block of 50000 tuples generated using F_2, F_3, and F_4. $D_{(1)}$=0.5M.F1, $D_{(2)}$=1M.F2, $D_{(3)}$=1M.F3, $D_{(4)}$=1M.F4, $D_{(5)}$=D+$\delta_{(5)}$=D + 0.05M.F2, $D_{(6)}$=D+$\delta_{(6)}$=D + 0.05M.F3, and $D_{(7)}$=D+$\delta_{(7)}$=D + 0.05M.F4.

The significance of the deviation for $D_{(1)}$ in row (1) is low because it has the same distribution as that of D. The significance of deviations in rows (2), (3), (4) are high, as expected. From rows (5), (6), (7), we see that even the addition of new blocks of size 50K to D causes significant deviations.

In Figure 14, we plot the misclassification error (ME) for the tree constructed from D with respect to a second dataset (chosen from $\delta_{(5)}$-$\delta_{(7)}$ and $D_{(2)} - D_{(4)}$) against the deviation between the two datasets. We see that they exhibit a strong positive correlation.

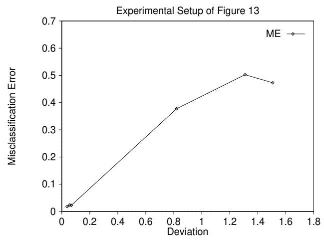

Figure 14: $Deviation$ vs. ME

8 Related and Future Work

A lot of research on clustering concentrated on detecting "outliers" within the dataset as noise and devised special strategies to handle them [34, 28, 15, 37, 22]. In contrast to the work on clustering, [24, 6, 26] concentrated primarily on discovering outliers in a dataset.

[35] proposed interestingness measures to monitor variation in a single pattern. [4, 10] addressed a similar problem of monitoring the support of an individual itemset. Given a pattern (or itemset) their algorithms propose to track its variation over a temporally ordered set of transactions. However, they do not detect variations at levels higher than that of a single pattern.

In future work, we intend to apply our framework to approximate query answering.

References

[1] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1998.

[2] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Database mining: A performance perspective. *IEEE Trans-*

Dataset	δ	% $sig(\delta)$	δ^*	Time for δ	Time for δ^*
$D_{(1)}$	0.0913	1	0.0913	0	0.01
$D_{(2)}$	3.2198	99	3.6893	46.27	0.01
$D_{(3)}$	6.0957	99	6.60874	46.16	0.01
$D_{(4)}$	6.0096	99	6.4435	44.19	0.01
$D+\delta_{(5)}$	0.1511	2	0.1610	17.37	0.0
$D+\delta_{(6)}$	0.2760	99	0.3645	19.53	0.01
$D+\delta_{(7)}$	0.2784	99	0.3668	18.86	0.0

Figure 12: Deviation with D: 1M.20L.1I.4pats.4plen

ID	δ	% $sig(\delta)$
$D_{(1)}$	0.0022	10
$D_{(2)}$	1.2068	99
$D_{(3)}$	0.8146	99
$D_{(4)}$	1.4819	99
$D+\delta_{(5)}$	0.0569	99
$D+\delta_{(6)}$	0.03722	99
$D+\delta_{(7)}$	0.0689	99

Figure 13: Deviation with D: 1M.F1

actions on Knowledge and Data Engineering, 5(6):914–925, December 1993.

[3] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast Discovery of Association Rules. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, chapter 12, pages 307–328. AAAI/MIT Press, 1996.

[4] Rakesh Agrawal and Giuseppe Psaila. Acive data mining. *Proceedings of the first international conference on knowledge discovery and data mining*, 1995.

[5] Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules. In *Proc. of the 20th Int'l Conference on Very Large Databases*, Santiago, Chile, September 1994.

[6] Andreas Arning, Rakesh Agrawal, and Prabhakar Raghavan. A linear method for deviation detection in large databases. In *Proc. of the 2nd Int'l Conference on Knowledge Discovery in Databases and Data Mining*, Portland, Oregon, August 1996.

[7] Peter J. Bickel and Kjell A. Doksum. *Mathematical Statistics: Basic Ideas and Selected Topics*. Prentice Hall, 1976.

[8] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, 1984.

[9] Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proc. of the ACM SIGMOD Conference on Management of Data*, May 1997.

[10] Soumen Chakrabarti, Sunita Sarawagi, and Byron Dom. Mining surprising patterns using temporal description length. In *Proceedings of the 24th International Conference on Very Large Databases*, pages 606–617, August 1998.

[11] D. Cheung, J. Han, V. Ng, and C.Y. Wong. Maintenance of discovered association rules in large databases: An incremental updating techniques. In *Proc. of 1996 Int'l Conference on Data Engineering*, New Orleans, USA, February 1996.

[12] E. F. Codd. Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate. Technical report, E. F. Codd and Associates, 1993.

[13] Ralph B. D'Agostino and Michael A. Stephens. *Goodness-of-fit techniques*. New York: M.Dekker, 1986.

[14] Bradley Efron and Robert J. Tibshirani. *An introduction to the bootstrap*. Chapman & Hall, 1993.

[15] Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. A database interface for clustering in large spatial databases. In *Proc. of the 1st Int'l Conference on Knowledge Discovery in Databases and Data Mining*, Montreal, Canada, August 1995.

[16] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.

[17] Ronen Feldman, Yonatan Aumann, Amihood Amir, and Heikki Mannila. Efficient algorithms for discovering frequent sets in incremental databases. *Workshop on Research issues on Data Mining and Knowledge Discovery*, 1997.

[18] Venkatesh Ganti, Raghu Ramakrishnan, Johannes Gehrke, Allison Powell, and James French. Clustering large datasets in arbitrary metric spaces. In *To appear in Proceedings of the IEEE International Conference on Data Engineering*, Sydney, March 1999.

[19] Johannes Gehrke, Raghu Ramakrishnan, and Venkatesh Ganti. Rainforest - a framework for fast decision tree construction of large datasets. In Ashish Gupta, Oded Shmueli, and Jennifer Widom, editors, *Proceedings of the 24th International Conference on Very Large Databases*, pages 416–427, New York, New York, August 1998. Morgan Kaufmann.

[20] David Gibson, Jon Kleinberg, and Prabhakar Raghavan. Clustering categorical data: An approach based on dynamical systems. In *Proceedings of the 24th International Conference on Very Large Databases*, pages 311–323, New York City, New York, August 24-27 1998.

[21] George Gratzer. *Lattice Theory: First Concepts and Distributive Lattices*. W. H. Freemand and Company, 1970.

[22] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure: An efficient clustering algorithm for large databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, June 1998.

[23] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. In *To appear in Proceedings of the IEEE International Conference on Data Engineering*, Sydney, March 1999.

[24] Isabelle Guyon, Nada Matic, and Vladmir Vapnik. Discovering informative patterns and data cleaning. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 181–204. AAAI Press, 1996.

[25] T. Imielinski and Heikki Mannila. A database perspective on knowledge discovery. *Communication of the ACM*, 39(11):58–64, Nov 1996.

[26] Edwin M. Knorr and Raymond T. Ng. Algorithms for distance-based outliers in large databases. In *Proceedings of the 24th International Conference on Very Large Databases*, pages 392–403, August 1998.

[27] Manish Mehta, Rakesh Agrawal, and Jorma Rissanen. SLIQ: A fast scalable classifier for data mining. In *Proc. of the Fifth Int'l Conference on Extending Database Technology (EDBT)*, Avignon, France, March 1996.

[28] Raymond T. Ng and Jiawei Han. Efficient and effective clustering methods for spatial data mining. In *Proc. of the VLDB Conference*, Santiago, Chile, September 1994.

[29] Raymond T. Ng, Laks V.S. Lakshmanan, Jaiwei Han, and Alex Pang. Exploratory mining and pruning optimizations of constrained association rules. In Laura Hass and Ashutosh Tiwary, editors, *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 13–24, June 1998.

[30] Jong Soo Park, Ming-Syan Chen, and Philip S. Yu. An effective hash based algorithm for mining association rules. In *Proc. of the ACM-SIGMOD Conference on Management of Data*, San Jose, California, May 1995.

[31] Rajeev Rastogi and Kyuseok Shim. Public: A decision tree classifier that integrates building and pruning. In *Proceedings of the 24th International Conference on Very Large Databases*, pages 404–415, New York City, New York, August 24-27 1998.

[32] A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. In *Proc. of the VLDB Conference*, Zurich, Switzerland, September 1995.

[33] John Shafer, Rakesh Agrawal, and Manish Mehta. SPRINT: A scalable parallel classifier for data mining. In *Proc. of the 22nd Int'l Conference on Very Large Databases*, Bombay, India, September 1996.

[34] J. W. Shavlik and T.G. Dietterich. *Readings in Machine Learning*. Morgan Kaufmann, 1990.

[35] Avi Silbershatz and Alexander Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(6), 1996.

[36] Shiby Thomas, Sreenath Bodagala, Khaled Alsabti, and Sanjay Ranka. An efficient algorithm for the incremental updation of association rules in large databases. In *Proceedings of 3rd International Conference on Knowledge Discovery in Databases*, 1997.

[37] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. BIRCH: An efficient data clustering method for very large databases. In *Proc. of the ACM SIGMOD Conference on Management of Data*, Montreal, Canada, June 1996.

A Preliminaries

In this section, we define three widely studied data mining models in the database literature: the *set of frequent itemsets*, the *decision tree classifiers*, and the *set of clusters*.

Set of Frequent Itemsets: Let $\mathcal{I} = \{i_1, \ldots, i_n\}$ be a set of literals called *items*. A *transaction* and an *itemset* are subsets of $\mathcal{I}$. A transaction t is said to *contain* the itemset X if $X \subseteq t$. Let D be a set of transactions. The *support* $sup_D(X)$ of an itemset X in D is the fraction of the total number of transactions in D that contain X: $sup_D(X) \stackrel{\text{def}}{=} \frac{|\{t: t \in D: X \subseteq t\}|}{|D|}$. We write $sup(X)$ instead of $sup_D(X)$ if D is clear from the context.

Let $m_s, 0 < m_s < 1$ be a constant called the *minimum support*. An itemset X is said to be *frequent* on D if $sup(X) \geq m_s$. The set of *frequent itemsets* $L_D^{m_s}$ consists of all itemsets that are frequent on D; formally, $L_D^{m_s} = \{X : X \subset \mathcal{I}, sup(X) \geq m_s\}$.

We now show how the model of frequent itemsets can be cast in the notation of Section 2. Each item i corresponds to an attribute whose domain is $\mathcal{D}_i = \{0, 1\}$. (Comparing with the description in the previous paragraph, the attribute space is the powerset $\mathcal{P}(\mathcal{I})$ of $\mathcal{I}$.) A transaction t can be expressed as an n-dimensional vector $t^n = \langle t_1, \ldots, t_n \rangle$ where t_j is 1 if $\{i_j\} \subseteq t$ and 0 otherwise for $j \in \{1, \ldots, n\}$. Thus the dataset D can be viewed as a set of n-tuples. Each itemset X represents the region: $\gamma_X = \mathcal{V}_1 \times \cdots \times \mathcal{V}_n$ where $\mathcal{V}_i = \{1\}$, if X contains item i, and $\mathcal{V}_i = \{0, 1\}$, otherwise. (If $i \notin X$, we set $\mathcal{V}_i = \{0, 1\}$ because a transaction t may support (or equivalently be mapped into) γ_X, even if it contains items that are not in X.) The support $sup(X)$ is the selectivity $\sigma(\gamma_X, D)$ of γ_X with respect to D.

Decision Tree Classifier: Let D be a dataset where each tuple has n attributes, some of which are numeric and some categorical[7]. There is one distinguished categorical attribute C called the *class label*, the remaining attributes are called *predictor attributes*. (We will denote the elements of the domain of the class label attribute as *class labels*; the semantics of the term class label will be clear from the context.) A decision tree T is a model of the dataset D that encodes the distribution of the class label in terms of the predictor attributes. It is a directed, acyclic graph in the form of a tree. The root of the tree does not have any incoming edges. All other nodes have exactly one incoming

[7] Attributes whose domain is totally ordered are called *numeric* attributes, whereas attributes whose domain is not ordered are called *categorical* attributes.

edge and either zero or more outgoing edges. If a node n has no outgoing edges then n is called a *leaf node*, otherwise it is called an *internal node*. Each leaf node n is labeled with one class label n_C. Each edge e originating from an internal node n has a predicate q associated with it, where q involves only predictor attributes. The set of predicates P on the outgoing edges of an internal node must be *non-overlapping* and *exhaustive*. A set of predicates P is *non-overlapping* if the conjunction of any two predicates in P evaluates to `false`. A set of predicates P is *exhaustive* if the disjunction of all predicates in P evaluates to `true`. Thus each leaf node n has associated a predicate P_n where P_n is the conjunction of the predicates along the edges on the path from the root node to n. Each leaf node encodes a classification rule $P_n \to n_C$, where n_C is the class label of n. Informally, if $P_n(t) = \texttt{true}$, then T predicts that t has class label n_C.

Without loss of generality, let $\mathcal{I} = \{A_1, \ldots, A_{n-1}, C\}$; the domain of predictor attribute A_i is $\mathcal{D}_i$, the domain of the class label attribute C is $\{1, \ldots, k\}$. The attribute space is $\mathcal{A}(\mathcal{I}) = \mathcal{D}_1 \times \ldots \times \mathcal{D}_{n-1} \times \{1, \ldots, k\}$. For each leaf node n, the set of predicates $\{P_n \wedge (C = i)\}$, $i = 1, \ldots, k$ defines a set of regions $\{\gamma_1^n, \ldots, \gamma_k^n\}$. The collection $\Gamma = \{\gamma_1^n, \ldots, \gamma_k^n : n \text{ is a leaf node in } T\}$ of regions is *non-overlapping* and *exhaustive* thus inducing a *partition* on $\mathcal{A}(\mathcal{I})$. We refer to Γ as the set of regions associated with T.

Clustering: The clustering problem has been widely studied and several definitions for a cluster have been proposed to suit the target applications. We define the set of clusters as the set of interesting regions discovered by a clustering algorithm. A set of regions is interesting if it satisfies a *cluster condition*, usually based on the density of points or distances between them.

B Proofs of Results

In this section, we give proofs of all claims made in the paper. The proofs are ordered by the class of models and thus do not appear in the same order as in the main part of the paper.

B.1 Proofs for lits-models

Proposition 4.1: The class of lits-models $\mathcal{M}$ on the set of items $\mathcal{I}$ exhibits the 2-component property and $\langle \Gamma_{\mathcal{M}}; \preceq_L \rangle$ is a meet-semilattice.

Proof: That the class of lits-models exhibits the 2-component property was illustrated in Section 4.1. The meet-semilattice property follows from the existence of the unique greatest common refinement under the refinement relation $\preceq_L$: $\Gamma_{\bigwedge(M_1,M_2)} = L_{D_1}^{m_s} \cup L_{D_2}^{m_s}$ is unique. $\square$

Lemma 4.1: Let $f \in \{f_a, f_s\}$ and $g \in \{g_{sum}, g_{max}\}$. Let Γ_M be a common refinement of Γ_{M_1} and Γ_{M_2}. Then,

$$\delta(M_1, M_2) \leq \delta^1_{(f,g)}(\langle \Gamma_M, \Sigma(\Gamma_M, D_1)\rangle, \langle \Gamma_M, \Sigma(\Gamma_M, D_2)\rangle)$$

Sketch of proof: Let $\delta'(M_1, M_2) = \delta^1_{(f,g)}(\langle \Gamma_M, \Sigma(\Gamma_M, D_1)\rangle, \langle \Gamma_M, \Sigma(\Gamma_M, D_2)\rangle)$. The result follows from these observations:

(1) $\Gamma_M \supseteq \Gamma_{\bigwedge(M_1,M_2)}$

(2) f is a non-negative function

(3) Let S_1 and S_2 be sets of positive numbers. Since $g \in \{g_{sum}, g_{max}\}$ it follows that $S_1 \subseteq S_2 \Rightarrow g(S_1) \leq g(S_2)$. Let X be an itemset and let $\kappa_1^X = sup_{D_1}(X) \cdot |D_1|$ and $\kappa_2^X = sup_{D_2}(X) \cdot |D_2|$. Then the following equation shows that Lemma 4.1 holds:

$$\begin{aligned}
&\delta'_{(f,g)}(M_1, M_2) \\
= \;& g(\{f(\kappa_1^X, \kappa_2^X, |D_1|, |D_2|) : X \in \Gamma_M\}) \\
= \;& g(\{f(\kappa_1^X, \kappa_2^X, |D_1|, |D_2|) : X \in \Gamma_{\bigwedge(M_1,M_2)}\})\} \cup \\
& \quad \{f(\kappa_1^X, \kappa_2^X, |D_1|, |D_2|) : X \in \Gamma_M - \Gamma_{\bigwedge(M_1,M_2)}\})\}) \\
\geq \;& g(\{f(\kappa_1^X, \kappa_2^X, |D_1|, |D_2|) : X \in \Gamma_{\bigwedge(M_1,M_2)}\}) \\
= \;& \delta_{(f,g)}(M_1, M_2)
\end{aligned}$$

$\square$

Theorem 4.2: Let $M_1, M_2 \in \mathcal{M}$ be two models induced by D_1, D_2 and let $g \in \{g_{sum}, g_{max}\}$. Then the following properties hold:

`(1)` $\delta^*_{(g)}(M_1, M_2) \geq \delta_{(f_a,g)}(M_1, M_2)$

`(2)` $\delta^*_{(g)}$ satisfies the triangular inequality.

`(3)` $\delta^*_{(g)}$ can be computed without scanning D_1 or D_2.

Sketch of proof: The GCR of two sets of frequent itemsets $L_{D_1}^{m_s}$ and $L_{D_2}^{m_s}$ is given by their union. Let $X \in L_{D_1}^{m_s} \cup L_{D_2}^{m_s}$. Let $\kappa_1^X = sup_{D_1}(X) \cdot |D_1|$ and $\kappa_2^X = sup_{D_2}(X) \cdot |D_2|$. Then exactly one of the following three cases holds.

`(C1)` $X \in L_{D_1}^{m_s} \cap L_{D_2}^{m_s}$: $sup(X)$ is known for D_1 and D_2.
$f^*(\kappa_1^X, \kappa_2^X, |D_1|, |D_2|) = f_a(\kappa_1^X, \kappa_2^X, |D_1|, |D_2|)$

`(C2)` $X \in L_{D_1}^{m_s} - L_{D_2}^{m_s}$: $sup(X)$ is known for D_1 only.
$f^*(\kappa_1^X, \kappa_2^X, |D_1|, |D_2|) \geq f_a(\kappa_1^X, 0, |D_1|, |D_2|)$

`(C3)` $X \in L_{D_2}^{m_s} - L_{D_1}^{m_s}$: $sup(X)$ is known for D_2 only.
$f^*(\kappa_1^X, \kappa_2^X, |D_1|, |D_2|) \geq f_a(0, \kappa_2^X, |D_1|, |D_2|)$

From the above three observations and since $g \in \{g_{sum}, g_{max}\}$, `(1)` holds. Since the computation of δ^* requires only information from the measure components of the two models M_1 and M_2, `(3)` is true.

We now prove property `(2)`. Let D_3 be another dataset which induces the model M_3. We partition $L_{D_1}^{m_s} \cup L_{D_2}^{m_s} \cup L_{D_3}^{m_s}$ into seven disjoint regions `(A)-(G)` as shown in Figure 15. We now show that

$$\delta^*_{(g)}(M_1, M_2) + \delta^*_{(g)}(M_2, M_3) \geq \delta^*_{(g)}(M_1, M_3).$$

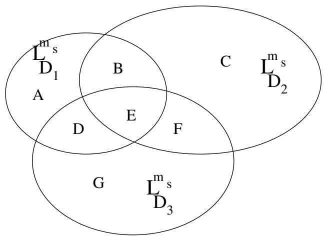

Figure 15: Three sets of frequent itemsets: $L^{m_s}_{D_1}, L^{m_s}_{D_2}, L^{m_s}_{D_3}$

$\delta^*_{(g)}(M_1, M_3) =$

$$\sum_{X\in A\cup B} sup_{D_1}(X) + \sum_{X\in D} |sup_{D_1}(X) - sup_{D_3}(X)|$$
$$+ \sum_{X\in E} |sup_{D_1}(X) - sup_{D_3}(X)| + \sum_{X\in F\cup G} sup_{D_3}(X)$$

$\delta^*_{(g)}(M_1, M_2) + \delta^*(g)(M_2, M_3) =$
$\sum_{X\in A} sup_{D_1}(X) + \sum_{X\in C} sup_{D_2}(X) + \sum_{X\in G} sup_{D_3}(X)+$
$\sum_{X\in B}(|sup_{D_1}(X) - sup_{D_2}(X)| + sup_{D_2}(X))+$
$\sum_{X\in D}(|sup_{D_1}(X) - sup_{D_3}(X)| + sup_{D_3}(X))+$
$\sum_{X\in E}(|sup_{D_1}(X) - sup_{D_2}(X)| + |sup_{D_2}(X) - sup_{D_3}(X)|)+$
$\sum_{X\in F}(|sup_{D_2}(X) - sup_{D_3}(X)| + sup_{D_3}(X))$

The result follows from the following inequalities. Let $X \in L^{m_s}_{D_1} \cup L^{m_s}_{D_2} \cup L^{m_s}_{D_3}$ be an itemset. Then,

$X \in B \Rightarrow$
$|sup_{D_1}(X) - sup_{D_2}(X)| + sup_{D_2}(X) \geq sup_{D_1}(X)$
$X \in D \Rightarrow$
$|sup_{D_1}(X) - sup_{D_3}(X)| + sup_{D_3}(X)$
$\geq |sup_{D_1}(X) - sup_{D_3}(X)|$
$X \in E \Rightarrow$
$|sup_{D_1}(X) - sup_{D_2}(X)| + |sup_{D_2}(X) - sup_{D_3}(X)|$
$\geq |sup_{D_1}(X) - sup_{D_3}(X)|$
$X \in F \Rightarrow$
$|sup_{D_2}(X) - sup_{D_3}(X)| + sup_{D_3}(X) \geq sup_{D_2}(X)$

□

B.2 Proofs for dt-models

Proposition 4.2: Let $\mathcal{M}$ be the class of dt-models with refinement relation $\preceq_T$. Then $\mathcal{M}$ exhibits the 2-component property and $\langle \Gamma_{\mathcal{M}}; \preceq_T \rangle$ is a meet-semilattice.
Proof: The 2-component property has already been discussed in Section 4.2. The meet-semilattice property follows from the uniqueness of the GCR, which we prove by showing that if

$$(\Gamma_{M'} \preceq_T \Gamma_{M_1}) \wedge (\Gamma_{M'} \preceq_T \Gamma_{M_2}) \Rightarrow (strucM' \preceq_T \Gamma_{\bigwedge(M_1,M_2)}$$

Let $M_1, M_2 \in \mathcal{M}$ be induced by two datasets D_1, D_2. Let $\Gamma_M = \Gamma_{\bigwedge(M_1,M_2)}$ be defined as follows: $\Gamma_{\bigwedge(M_1,M_2)} \stackrel{\text{def}}{=}$ $\{\gamma : \gamma$ is identified by $P_{\gamma_1} \wedge P_{\gamma_2} \ni \gamma_1 \in \Gamma_{M_1} \wedge \gamma_2 \in \Gamma_{M_2}\}$

Let $\Gamma_{M'}$ also refine Γ_{M_1} and Γ_{M_2}. Let $\gamma \in \Gamma_{M'}$. Then, there exist $\gamma_1 \in \Gamma_{M_1}$ and $\gamma_2 \in \Gamma_{M_2}$ such that γ is a sub-region of both γ_1 and γ_2. Therefore, γ is a sub-region of $\gamma_{12} \in \Gamma_M$ identified by the predicate $P_{\gamma_1} \wedge P_{\gamma_2}$. Similarly, every region in $\Gamma_{M'}$ is a sub-region of some region in Γ_M. Since $\Gamma_{M'}$ is a partition of the attribute space every region in Γ_M is refined by a set of regions in $\Gamma_{M'}$. Therefore, $\Gamma_{M'} \preceq_T \Gamma_M$. □

Theorem 4.3: Let Γ_M be a common refinement of Γ_{M_1} and Γ_{M_2}. Let $g = g_{sum}$, and $f \in \{f_a, f_s\}$. Then, $\delta_{(f,g_{sum})}(M_1, M_2) \leq$
$\delta^1_{(f,g)}(\langle \Gamma_M, \Sigma(\Gamma_M, D_1)\rangle, \langle \Gamma_M, \Sigma(\Gamma_M, D_2)\rangle)$
Proof: Let $\gamma \in \Gamma_{\bigwedge(M_1,M_2)}$ be a region. Let x and y be the number of tuples in D_1 and D_2 respectively that are mapped into γ. Since Γ_M refines $\Gamma_{\bigwedge(M_1,M_2)}$, there exists a set of regions $\gamma_1, \ldots, \gamma_k \in \Gamma_M$ such that they partition γ. Then, x is split into $x_1, \ldots, x_k$ at $\gamma_1, \ldots, \gamma_k$ respectively such that $(x_1 + \ldots + x_k) = x$. Similarly, y is split into $y_1, \ldots, y_k$ such that $y = y_1 + \ldots + y_k$. Therefore, the computation of δ' further splits the term $f(x, y, |D_1|, |D_2|)$ into $\sum_{i=1}^k f(x_i, y_i, |D_1|, |D_2|)$. The computation of δ sums $f(x, y, |D_1|, |D_2|)$ at each region γ in $\Gamma_{\bigwedge(M_1,M_2)}$. We consider the relationship between δ and δ' at an arbitrary region $\gamma \in \Gamma_{\bigwedge(M_1,M_2)}$, which extends to all of $\Gamma_{\bigwedge(M_1,M_2)}$.
`Case 1:` $f = f_a$.

$$\begin{aligned} f_a(x, y, |D_1|, |D_2|) &= \left|\frac{x}{|D_1|} - \frac{y}{|D_2|}\right| \\ &= \left|\sum_{i=1}^k \left(\frac{x_i}{|D_1|} - \frac{y_i}{|D_2|}\right)\right| \\ &\leq \sum_{i=1}^k \left|\frac{x_i}{|D_1|} - \frac{y_i}{|D_2|}\right| \\ &= \sum_{i=1}^k f_a(x_i, y_i, |D_1|, |D_2|) \end{aligned}$$

`Case 2:` $f = f_s$.

$$\begin{aligned} f_s(x, y, |D_1|, |D_2|) &= \frac{|\frac{x}{|D_1|} - \frac{y}{|D_2|}|}{\frac{1}{2}(\frac{x}{|D_1|} + \frac{y}{|D_2|})} \\ &\leq \frac{\sum_{i=1}^k |\frac{x_i}{|D_1|} - \frac{y_i}{|D_2|}|}{\frac{1}{2}(\frac{x}{|D_1|} + \frac{y}{|D_2|})} \\ &\leq \sum_{i=1}^k \frac{|\frac{x_i}{|D_1|} - \frac{y_i}{|D_2|}|}{\frac{1}{2}(\frac{x_i}{|D_1|} + \frac{y_i}{|D_2|})} \\ &= \sum_{i=1}^k f_s(x_i, y_i, |D_1|, |D_2|) \end{aligned}$$

Since γ is arbitrarily chosen, the relationship extends to the entire structural component of $\Gamma_{\bigwedge(M_1,M_2)}$ in a straightforward way. □

Counter Example for $g = g_{max}$:
Using the above notation, let $x = 0.5$, $y = 0.0$ be the measures from D_1 and D_2 respectively at a region γ. Let $x_1 = x_2 = 0.25$ and $y_1 = y_2 = 0.0$ be the measures of the partition of γ into $\{\gamma_1, \gamma_2\}$. If we use g_{max}, clearly the partition may lead to a lower value for δ. Therefore, the above lemma is not valid for g_{max}.

Theorem 5.2: Let T be a dt-model induced by D_1. Let D_2 be another dataset. Then $ME^T(D_2)$=

$$\frac{1}{2}\delta_{(f_a, g_{sum})}(\langle \Gamma_T, \Sigma(\Gamma_T, D_2)\rangle, \langle \Gamma_T, \Sigma(\Gamma_T, D_2^T)\rangle)$$

Proof: Let n be a leaf node in T associated with class label n_C, predicate P_n, and regions $\gamma_i^n = P_n \wedge (C = i)$, $1 \le i \le k$. Let $x_i, 1 \le i \le k$ be the number of tuples in D_2 that satisfy the predicate P_n and belong to the class i. Let $x = x_1 + \cdots x_k$. Without loss of generality assume that $n_C = 1$. The misclassification error of T at n is $\frac{x - x_1}{|D_2|}$. Since T predicts that all tuples in D_2 satisfying P_n are in class 1, the number of tuples in D_2^T at regions $\gamma_1^n, \ldots, \gamma_k^n$ are $(x_1 + \ldots + x_k), 0, \ldots, 0$ respectively. Aggregating the differences between the measures in D_2 and in D_2^T over regions $\gamma_1^n, \ldots, \gamma_k^n$ and noting that $|D_2| = |D_2^T|$, we get: $f_a(x_1, x_1 + \ldots + x_k, |D_2|, |D_2|) + \sum_{i=2}^{k} f_a(x_i, 0, |D_2|, |D_2|)$
$= \frac{x - x_1}{|D_2|} + \sum_{i=2}^{k} \frac{x_i}{|D_2|} = \frac{2(x - x_1)}{|D_2|}$
Summation over all leaf nodes results in $2ME^T(D_2)$. □

Proposition 5.1:
Let T be the decision tree induced by D_1, and let D_2 be another dataset. Then the chi-squared statistic X^2 is given by:

$$X^2 = \delta_{(f, g_{sum})}(\langle T, \Sigma(T, D_1)\rangle, \langle T, \Sigma(T, D_2)\rangle) \text{ where}$$

$$f(v_1, v_2, |D_1|, |D_2|) = \begin{cases} \frac{|D_2|\left(\frac{v_1}{|D_1|} - \frac{v_2}{|D_2|}\right)^2}{\frac{v_1}{|D_1|}}, & \text{if } v_1 > 0 \\ c, & \text{otherwise} \end{cases}$$

Sketch of proof: The result follows from the definitions of f and the chi-squared statistic. □

B.3 Focussed Deviations

Theorem 5.1: Let $\mathcal{M}$ be one of the following three classes of models: lits-models, dt-models, cluster-models. Let $\preceq$ be a refinement relation such that $\langle \Gamma_{\mathcal{M}}; \preceq\rangle$ forms a meet-semilattice. Let $\gamma \subseteq \mathcal{A}(\mathcal{I})$ be the focussing region. Then $\langle \Gamma_{\mathcal{M}}^{\gamma}; \preceq\rangle$ is a meet-semilattice.
Sketch of proof: Let $M_1, M_2 \in \mathcal{M}$. The result follows from the following two observations.

First, if $\Gamma_{M_1} \preceq \Gamma_{M_2}$ then $\Gamma_{M_1}^{\gamma} \preceq \Gamma_{M_2}^{\gamma}$. Because, $\{\gamma_{M_1}^{j_1}, \ldots, \gamma_{M_1}^{j_k}\}$ refines $\gamma_{M_2}^{i}$ then $\{\gamma \cap \gamma_{M_1}^{j_1}, \ldots, \gamma \cap \gamma_{M_1}^{j_k}\}$ refines $\gamma \cap \gamma_{M_2}^{i}$.

Second, focussing (with respect to γ) the GCR of two structural components is identical to the GCR of the focussed structural components, i.e., $\Gamma^{\gamma}_{\bigwedge(M_1, M_2)} = \bigwedge(\Gamma_{M_1}^{\gamma}, \Gamma_{M_2}^{\gamma})$. We prove this observation for lits-models and dt-models separately.
lits-models: $\Gamma^{\gamma}_{\bigwedge(M_1, M_2)}$

$$\begin{aligned} &= \gamma \cap \Big\{\{\gamma_{11}, \ldots, \gamma_{1|\Gamma_{M_1}}\} \cup \{\gamma_{21}, \ldots, \gamma_{2|\Gamma_{M_2}}\}\Big\} \\ &= \Big\{\{\gamma \cap \gamma_{11}, \ldots, \gamma \cap \gamma_{1|\Gamma_{M_1}}\} \cup \\ &\qquad \{\gamma \cap \gamma_{21}, \ldots, \gamma \cap \gamma_{2|\Gamma_{M_2}}\}\Big\} \\ &= \bigwedge(\Gamma_{M_1}^{\gamma}, \Gamma_{M_2}^{\gamma}) \end{aligned}$$

dt-models: We use the predicate notation for regions in a dt-model. $\Gamma^{\gamma}_{\bigwedge(M_1, M_2)}$

$$\begin{aligned} &= P_{\gamma} \wedge \{P_{\gamma_1} \wedge P_{\gamma_2} \ni \gamma_1 \in \Gamma_{M_1} \wedge \gamma_2 \in \Gamma_{M_2}\} \\ &= \{(P_{\gamma} \wedge P_{\gamma_1}) \wedge (P_{\gamma} \wedge P_{\gamma_2}) \ni \gamma_1 \in \Gamma_{M_1} \wedge \gamma_2 \in \Gamma_{M_2}\} \\ &= \bigwedge(\Gamma_{M_1}^{\gamma}, \Gamma_{M_2}^{\gamma}) \end{aligned}$$

Therefore, $\langle \Gamma_{\mathcal{M}}^{\gamma}; \preceq\rangle$ forms a meet-semilattice. □

C The Qualification Procedure

Suppose the deviation (obtained from an instantiation of FOCUS) between D_1 and D_2 is 0.005, and that between D_1 and D_3 is 0.01. From just the deviations, we are able to say the data characteristics of D_1 and D_2 are more similar than those of D_1 and D_3. But, we still do not know whether they have "different" data characteristics because a deviation of 0.005 may not be uncommon between two datasets generated by the same process. The qualification procedure addresses this question and returns the "statistical significance" (or confidence) of the deviation, i.e., whether or not D_1 and D_2 are generated by the same generating process with, say, 95% probability.

Let M_1 and M_2 be two models induced by D_1 and D_2 respectively. Let $\mathcal{G}$ be the hypothetical process that generated D_1. Let us assume that we know the distribution $\mathcal{F}$ of deviations between models induced by a randomly selected pair of datasets generated by $\mathcal{G}$. (We will describe a procedure to compute $\mathcal{F}$ in the next paragraph.) If D_1 and D_2 have the same data characteristics (equivalently, if D_2 was also generated by $\mathcal{G}$), then $d = \delta(M_1, M_2)$ would be a value drawn from $\mathcal{F}$. Since we know $\mathcal{F}$, we compute the probability p_d (=$P(F > d)$ where F is a random variable with distribution $\mathcal{F}$) of two models induced by datasets generated by $\mathcal{G}$ having a deviation greater than or equal to d. If p_d is "significantly" low, then the probability that D_2 was also generated by $\mathcal{G}$ is very low, and we can conclude that D_1 and D_2 have different data characteristics with confidence $100(1 - p_d)\%$. Typically, most statistical methods consider a value less than `0.05` to be significantly low. $100(1-p_d)\%$ is the *percentage significance* of the deviation between D_1 and D_2.

Algorithm C.1 Qualify($D_1, M_1, D_2, M_2, \delta, m, n$)
/* M_1, M_2 are models induced by D_1, D_2. */
/* Output: $\hat{p}_\delta = P_0(F > \delta(M_1, M_2))$ */
begin
Draw samples $S_{11}, S_{12}, \ldots, S_{n1}, S_{n2}$ of size m each from D_1.
Compute $\Delta = \{\delta_1, \ldots, \delta_n\}$ where $\delta_i = \delta(M^S_{i1}, M^S_{i2})$ and M^S_{i1}, M^S_{i2} are induced by S_{i1}, S_{i2}.
Let $Delta_{(<)} = \{\delta_{(0)}, \delta_{(1)}, \ldots, \delta_{(n+1)}\}$ be the list of values in $\Delta \cup \{0, \infty\}$ sorted in increasing order.
Find i such that $\delta_{(i)} \leq \delta(M_1, M_2) < \delta_{(i+1)}$.
$\hat{p}_\delta = \frac{n+1-i}{n+1}$.
end

D Chi-squared Goodness of Fit Test

Let D be a dataset and M, a model to be tested for its goodness of fit against D. The chi-squared test assumes that the entire space is partitioned into cells $\Gamma = \{\gamma_1, \ldots, \gamma_m\}$. Each cell γ_i is associated with an *expected measure* $E[\gamma_i, D]$ and an *observed measure* $O[\gamma_i, D]$. $E[\gamma_i, D]$ is the number of tuples in D predicted to be mapped into γ_i, whereas $O[\gamma_i, D]$ is the actual number. Then the *chi-squared statistic* is computed as follows.

$$X^2 = \Sigma_{i=1}^{m} \frac{(O[\gamma_i, D] - E[\gamma_i, D])^2}{E[\gamma_i, D]}$$

The value of X^2 is then compared with the chi-squared tables to obtain the significance of M's fit to D. (See [13] for more details.) If X^2 is higher than a cutoff value (3.84 at 95% significance level) then we reject the hypothesis that M fits D well.

E Wilcoxon Two-sample test

Let s_i, s_{i+1} be two sample sizes being compared. Compute the sample deviation measures of m random samples each of size s_i drawn from D. Let the measures be $\Delta_i = \{\delta_{i,1}, \ldots, \delta_{i,m}\}$. Repeat the procedure for size s_{i+1} yielding the set $\Delta_{i+1} = \{\delta_{(i+1),1}, \ldots, \delta_{(i+1),m}\}$. Given the two sequences of values, $\Delta_{i+1} = \{\delta_{(i+1),1}, \ldots, \delta_{(i+1),m}\}$ and $\Delta_i = \{\delta_{i,1}, \ldots, \delta_{i,m}\}$, the Wilcoxon test checks if many values in Δ_{i+1} are smaller than Δ_i. If both Δ_{i+1} and Δ_i come from identical distributions then we expect half of all possible pairs $(x, y), x \in \Delta_{i+1}, y \in \Delta_i$ to have $x < y$ and the other half to have $x > y$. If not, the number of pairs (x, y) where $x < y$ is skewed. This hypothesis is captured by the test statistic W.

$$U = |\{(x, y) : x \in \Delta_{i+1}, y \in \Delta_i, x < y\}|$$

$$W = \frac{U - Mean(U)}{\sqrt{Var(U)}}$$

where $Mean(U) = \dfrac{m^2}{2}$ and $Var(U) = \dfrac{m^2(2m+1)}{12}$

W is Normally distributed [7]. Let $W = z_\alpha$ where for a normally distributed random variable Z with mean 0 and unit variance, $P(Z \leq z_\alpha) = \alpha$. Then the hypothesis that Δ_{i+1} and Δ_i have the same distribution can be rejected with $100(1-\alpha)\%$ confidence. (See [7] for more details.) The hypothesis that samples of size s_{i+1} is more representative than samples of size s_i has significance $100(1-\alpha)\%$.

F Representativeness of a Sample

In this section, we show additional plots of sample deviations versus sample fraction for combinations of difference function f and aggregate function g that could not be presented in the main part of the paper due to space constraints.

Figures 16, 17, and 18 show the plots of sample deviation versus sample fraction on the dataset `1M.20L.1I.4pats.4plen` for the following deviation functions: $\delta_{(f_a, g_{max})}$, $\delta_{(f_s, g_{sum})}$, and $\delta_{(f_s, g_{max})}$. We see that the conclusions drawn from the plot for $f = f_a, g = g_{sum}$ hold even for these deviation functions.

Figures 19, 20, and 21 show the plots of sample deviation value versus sample fraction on datasets generated using classification functions F1, F2, F3, and F4 for the following deviation functions: $\delta_{(f_a, g_{max})}$, $\delta_{(f_s, g_{sum})}$, and $\delta_{(f_s, g_{max})}$. In Figures 20 and 21, the sample deviation values for the classification function F2 are much smaller than the sample deviation values of the other classification functions. The reason for this difference is as follows. The structural component of the classification tree models constructed using datasets from F2 involve only categorical attributes. Due to the small domain of the categorical attributes, the exact structural component of the classification tree model for F2 is recognized even with a very small sample. Datasets from other classification functions induce classification tree models with numerical attributes in their structural component. Due to random fluctuations, the structural components of models induced by a sample differ slightly from the structural component of the model induced by the entire dataset. When $f = f_a$, this difference is very small (see Figure 19) whereas it is pronounced when f_s is used (see Figures 20 and 21). The plots involving f_s illustrate its sensitivity to even minor differences in the structural component between two decision tree models.

Why is the sensitivity of the deviation functions derived from f_s for decision tree models so high? Consider two decision tree models M_1 and M_2 induced by D_1 and D_2

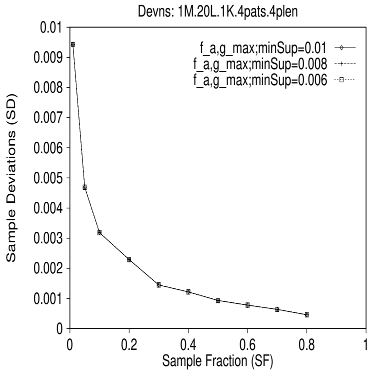

Figure 16: SD vs SF

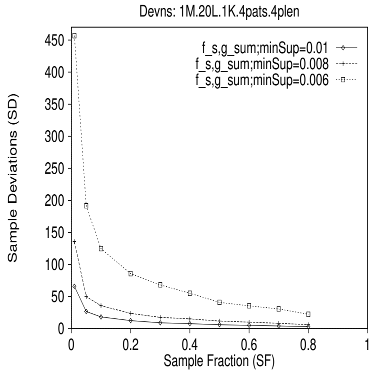

Figure 17: SD vs SF

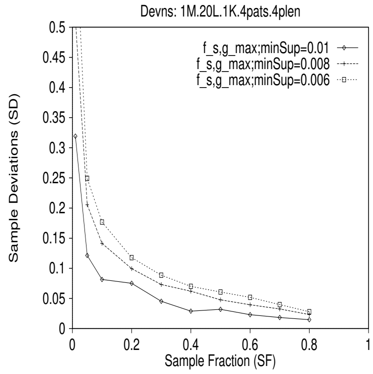

Figure 18: SD vs SF

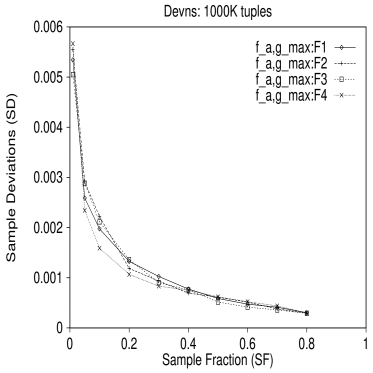

Figure 19: SD vs SF

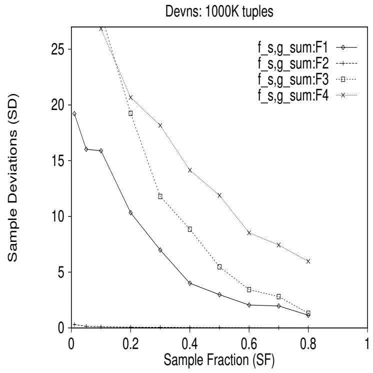

Figure 20: SD vs SF

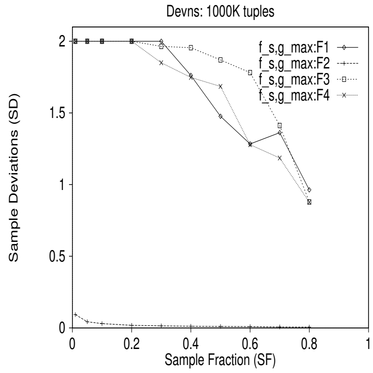

Figure 21: SD vs SF

respectively. Let γ be a region with measures v_1 in D_1 and v_2 in D_2. If the structural components Γ_{M_1} and Γ_{M_2} are different, it is possible that $v_1 > 0$ and $v_2 = 0$ in which case $f_s(v_1, v_2, |D_1|, |D_2|) = 2$. (Recall that the motivation behind f_s was to recognize such differences.) This case happens often as can be seen in Figure 20 and 21, because the decision tree model induced from the sample almost always has a slightly different structural component. When g_{max} is used, the sensitivity is extremely pronounced. Thus f_s is not a good function to be used when quantifying the representativeness of a sample. However, an interesting application for f_s is the detection of "exceptions" while monitoring changes.

The sensitivity is not as pronounced for frequent itemset models because an itemset X appears in the GCR of two models only if X has the required minimum support in at least the sample or the entire dataset. Thus the case described in the previous paragraph ($v_1 > 0$ and $v_2 = 0$) only happens extremely rarely.