# A Framework for Measuring Changes in Data Characteristics

Venkatesh Ganti Johannes Gehrke* Raghu Ramakrishnan†
{vganti,johannes,raghu}@cs.wisc.edu
Wei-Yin Loh‡
loh@stat.wisc.edu

**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 [1, 2, 3, 28, 29], for more efficient model construction [9, 15, 20, 21, 23, 31, 32, 33, 34, 38], and to deal with new data types [19, 21, 24]. There is, however, no work addressing the important issue of how to measure the difference, or *deviation*, between two models.

As a motivating example, consider the following 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, 37]. 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 (Section 5).
- The framework covers the models obtained by several mining algorithms, including frequent itemsets, decision trees, and clusters (Sections 4.1 and 4.2). 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

* Supported by an IBM Graduate Fellowship
† This research was supported by Grant 2053 from the IBM corporation.
‡ Supported by Army Research Office grant DAAG55-98-1-0333.

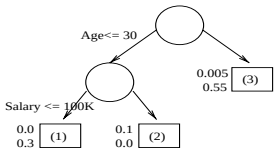



Figure 1: DT

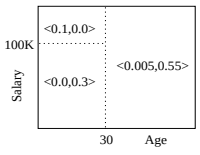
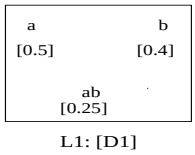



Figure 2: DT:2-components




L1: [D1]

Figure 3: Lits

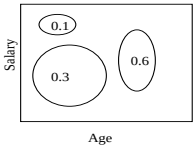



Figure 4: Clusters

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 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 Examples Illustrating Deviation

In general, a data mining model constructed from a dataset is designed to capture the interesting characteristics in the data. Therefore, we use the difference between data mining models as the measure of deviation between the underlying datasets. In this paper, we consider three classes of data mining models widely studied in the database literature: lits-models, dt-models, and cluster-models. 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. We assume that the reader is familiar with each of these classes of models. (For a formal description, see [3, 8, 38] or the full paper [18].) In this section, we illustrate the concepts and ideas behind the computation of deviation between two datasets first through the class of decision tree models and then through the class of frequent itemsets. In Section 3, we formalize these concepts.

### 2.1 dt-models

Let the decision tree constructed from a hypothetical dataset $D$ with two classes—$C_1$ and $C_2$—be as shown in Figure 1. The decision tree consists of three leaf nodes. The class distribution at each leaf node is shown beside it (on the left side) with the top (bottom) number denoting the fraction of database tuples that belong to class $C_1$ ($C_2$, respectively). For instance, the fractions of database tuples that belong to the classes $C_1$ and $C_2$ in the leaf node (1) are $0.0$ and $0.3$, respectively. Each leaf node in the decision tree corresponds to two

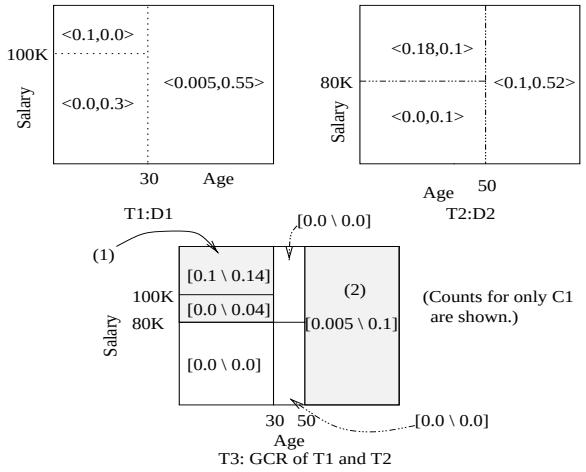



Figure 5: dt-model:$T_3 = \bigwedge(T_1, T_2)$

regions (one region for class $C_1$ and one region for class $C_2$), and each region is associated with the fraction of tuples in the dataset that map into it; this fraction is called the *measure* of the region. Generalizing from this example, each leaf node of a decision tree for $k$ classes is associated with $k$ regions in the attribute space each of which is associated with its measure. These $k$ regions differ only in the class label attribute. In fact, the set of regions associated with all the leaf nodes partition the attribute space.

We call the set of regions associated with all the leaf nodes in the dt-model the *structural component* of the model. We call the set of measures associated with each region in the structural component the *measure component* of the model. The property that a model consists of structural and measure components is called the 2-*component* property. Figure 2 shows the set of regions in the structural component of the decision tree in Figure 1 where the two regions corresponding to a leaf node are collapsed together for clarity in presentation. The two measures of a leaf node are shown as an ordered pair, e.g., the ordered pair $\langle 0.0, 0.3\rangle$ consists of the measures for the two collapsed regions of the leaf node (1) in Figure 1.

We now illustrate the idea behind the computation of deviation between two datasets over a set of regions. Let $D_1$ and $D_2$ be two datasets. Given a region and the measures of that region from the two datasets, the *deviation* between $D_1$ and $D_2$ w.r.t. the region is a function (e.g., absolute difference) of the two measures; we call this function the *difference function*. A generalization to the deviation over a set of regions is a "combination" of all their deviations at each region; we represent this combination of deviations by a function called the *aggregate function*, e.g., sum.

If two datasets $D_1$ and $D_2$ induce decision tree models with identical structural components, we can combine the two

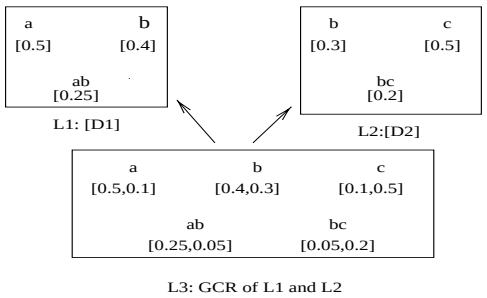



Figure 6: lits-model: $L_3 = \bigwedge(L_1, L_2)$

ideas—the 2-component property and the deviation w.r.t. a set of regions—to compute their deviation as follows: the deviation between $D_1$ and $D_2$ is the deviation between them w.r.t. the set of regions in their (identical) structural components.

However, the decision tree models induced by two distinct datasets typically have different structural components, and hence the simple strategy described above for computing deviations may not apply. Therefore, we first make their structural components identical by "extending" them. The extension operation relies on the structural relationships between models, and involves refining the two structural components by splitting regions until the two sets become identical. Intuitively, the refined set of regions is the finer partition obtained by overlaying the two partitions of the attribute space induced by the structural components of both decision trees. We call the refined set of regions the *greatest common refinement* (GCR) of the two structural components. For instance, in Figure 5, $T_3$ is the GCR of the two trees $T_1$ induced by $D_1$ and $T_2$ induced by $D_2$. In each region of the GCR $T_3$, we show a hypothetical set of measures (only for class $C_1$) from the datasets $D_1$ and $D_2$. For instance, the measures for the region $\texttt{salary} \geq 100K$ and $\texttt{age} < 30$ for the class $C_1$ from $D_1$ and $D_2$ are $0.0$ and $0.04$, respectively. The property that the GCR of two models always exists, which we establish later for decision tree models, is called the *meet-semilattice* property of the class of models.

To summarize, the deviation between two datasets $D_1$ and $D_2$ is computed as follows. The structural components of the two dt-models are extended to their GCR. Then, the deviation between $D_1$ and $D_2$ is the deviation between them over the set of all regions in the GCR. In Figure 5, if the difference function is the absolute difference and the aggregate function is the sum then (part of) the deviation between $D_1$ and $D_2$ over the set of all $C_1$ regions is given by the sum of deviations at each region in $T_3$: $|0.0-0.0|+|0.0-0.04|+|0.1-0.14|+|0.0-0.0|+|0.0-0.0|+|0.005-0.1| = 0.175$.

### 2.2 lits-models

Paralleling the above example computation using the class of decision tree models, we now illustrate the deviation computation through the class of frequent itemset models.

Figure 3 shows a simple itemset model where $\mathcal{I} = \{a, b\}$. It has three interesting regions identified by the frequent 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 $0.25$. The measure of a region identified by an itemset is the support of the itemset. Generalizing from this example, each frequent itemset $X$ in a lits-model represents a region in the attribute space (where the support is higher than the threshold) whose measure is the support of $X$. The set of all frequent itemsets is the *structural component* and the set of their supports is the *measure component*.

As in the case of decision trees, if the structural components of two models are identical we compute the deviation between them to be the aggregate of the deviations between the measures at all regions in either structural component. However, if the structural components are different, we first make them identical by extending both models to their *greatest common refinement*. For the lits-models, the GCR is the union of the sets of frequent itemsets of both models. For example, Figure 6 shows the GCR of two lits-models $L_1$ induced by $D_1$ and $L_2$ induced by $D_2$. Then, the deviation between the datasets is the deviation between them over the set of all regions in the GCR. The measures (or supports) from $D_1$ and $D_2$ for each itemset in the GCR are shown below it. If the difference function is the absolute difference, and the aggregate function is the sum then the deviation between $D_1$ and $D_2$ is $|0.5 - 0.1| + |0.4 - 0.3| + |0.1 - 0.5| + |0.25 - 0.05| + |0.05 - 0.2| = 1.125$.

### 2.3 Focussed Deviations

In the above examples, we computed the deviation between two datasets over the entire attribute space. In cases where an analyst is interactively exploring two datasets to find regions where they differ considerably, it is necessary to "focus" the deviation computation w.r.t. a specific region $R$. The FOCUS framework covers such requirements. The computation is focussed w.r.t. region $R$ by first intersecting each region in the GCR with $R$ and then combining (using the aggregate function) the deviations over these intersected regions. The intersection with $R$ ensures that the deviation is computed only over regions contained in $R$. In Figure 5, suppose the analyst is interested only in the difference between $T_1$ and $T_2$ over the region $R$: $\texttt{age} < 30$. The regions in the GCR $T_3$ intersected with $R$ are the three leftmost regions that satisfy the condition $\texttt{age} < 30$. The deviation between $T_1$ and $T_2$ w.r.t. $R$ is: $|0.0 - 0.0| + |0.0 - 0.04| + |0.1 - 0.14| = 0.08$.

A complementary approach is to declaratively specify a set of "interesting" regions in terms of the structural components of the two models and then rank the interesting regions in the order of their deviations. In Section 5, we introduce a set of structural operators and a ranking operator for declarative specification of interesting regions and region-ranking, respectively.

### 2.4 Additional Comments

A cluster-model induced by a dataset identifies a set of non-overlapping regions. Even though the set of regions in the structural component of a cluster-model may not be exhaustive, the discussion for cluster-models is a special case of dt-models. Due to space constraints, we do not discuss cluster-models in the rest of the paper.

Note that the derivation of the GCR of two models depends on the class of models being considered. We formalize this dependence in a later section. The computation of the deviation requires the measures from $D_1$ and $D_2$ over all the regions in the GCR to be computed; therefore, both the datasets need to be scanned once.

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 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"? We answer these questions rigorously using statistical techniques in Section 3.4.

We instantiate the misclassification error metric (from Machine Learning and Statistics) and the chi-squared goodness of fit statistic (from Statistics) from the FOCUS framework. Both metrics only consider the class of dt-models; thus, our FOCUS framework which covers other classes of models as well is more general than the current approaches in Machine Learning and Statistics.

## 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. In Section 4, we will prove that these properties are satisfied by lits-models, dt-models, and cluster-models.

### 3.1 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 $A_i$, $i \in \{1, \ldots, n\}$.

**Definition 3.1** 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* $\gamma$ 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 $\gamma$ has a corresponding predicate $P_\gamma$ such that $\{P_\gamma(t)$=`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 3.2** 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 map into $\gamma$: $\sigma(\gamma, D) \stackrel{\text{def}}{=} \frac{|\{t : t \in D \wedge t \in \gamma\}|}{|D|}$.

### 3.2 2-Component, Meet-semilattice Models

The main idea behind FOCUS is that a model $M$ has a structural component $\Gamma_M$ 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.

We already discussed the 2-component property of models in Section 2. We now describe the meet-semilattice property, which captures the structural relationship between models in a class of models $\mathcal{M}$. Figure 5 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 6, $L_3$ 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 components $\Gamma_{M_1}, \Gamma_{M_2}$ under the refinement relation always exists; we call this the *greatest common refinement (GCR)* of $\Gamma_{M_1}$ and $\Gamma_{M_2}$, 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 3.3** 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 $\Gamma_M = \{\gamma_M^i : 1 \leq i \leq l\}$ 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 3.4** 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 $\gamma_i$ if for any dataset $D$, $\sigma(\gamma_i, D) = \sum_{i=1}^{k} \sigma(\gamma_{j_i}, D)$. We say that $\Gamma_{M_1}$ *refines*

$\Gamma_{M_2}$ (denoted $\Gamma_{M_1} \preceq \Gamma_{M_2}$) if for every region $\gamma^j_{M_2} \in \Gamma_{M_2}$ there exists a set of regions $\{\gamma^{j_1}_{M_1}, \ldots, \gamma^{j_{k_j}}_{M_1}\} \subseteq \Gamma_{M_1}$ which refine $\gamma^j_{M_2}$. 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.

### 3.3 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 $\sigma(\gamma^i_{M_1}, D_1)$ and $\sigma(\gamma^i_{M_2}, D_2)$, $i = 1, \ldots, |\Gamma_{M_1}|$. 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 3.3.2. 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 $g : \mathcal{P}(\mathcal{R}_+) \mapsto \mathcal{R}_+$[1].

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

**Definition 3.5** 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 $\kappa^i_{D_j} = \sigma(\gamma_i, D_j) \cdot |D_j|$ denote the absolute number of tuples in $D_j$ that are mapped into $\gamma^i_{M_j} \in \Gamma_{M_j}$. 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 3.6** 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 $\delta$ 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 $g$.

#### 3.3.1 Computational Requirements for $\delta$

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.

#### 3.3.2 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 $\Gamma$. (Otherwise, we can extend them to their GCR.) Consider two itemsets $X_1$ and $X_2$ in $\Gamma$. Suppose $\sigma(\gamma^{X_1}_{L_1}, D_1) = 0.5, \sigma(\gamma^{X_1}_{L_2}, D_2) = 0.55$, and $\sigma(\gamma^{X_2}_{L_1}, D_1) = 0.0, \sigma(\gamma^{X_2}_{L_2}, D_2) = 0.05$. 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

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

supports, while the second instantiation $f_s$ "scales." We now define the two instantiations.[2]

**Definition 3.7** 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:

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

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

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 $\delta$.

### 3.4 The Qualification Procedure

Is the deviation sufficiently large that it is unlikely that the two datasets are generated by the same generating process? 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 $\mathcal{F}$. We omit the details due to space constraints. (See the full paper for details of the bootstrapping procedure and the statistical tests [18].)

## 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[3]. 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 [22]. (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.

Once again, consider the example in Figure 6. $L_3$ is the GCR of $L_1$ and $L_2$. The supports from $D_1$ and $D_2$ for each itemset in the GCR 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 $\Gamma_M$ be a common refinement of $\Gamma_{M_1}$ and $\Gamma_{M_2}$. Then,

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

#### 4.1.1 Upper Bound $\delta^*$ for $\delta$

In an exploratory, interactive environment where $\delta$ 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 $\delta^*$ of $\delta$ that can be computed fast using just the two models (which will probably fit in main memory, unlike the datasets). Using the upper bound $\delta^*$ instead of $\delta$ is safe; we will not ignore significant deviations. $\delta^*$ also satisfies the *triangle 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 triangle inequality.
(3) $\delta^*_{(g)}$ can be computed without scanning $D_1$ or $D_2$.

[2] 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.

[3] $L_{D_1}^{m_s}$ is the set of itemsets in $D_1$ with support greater than $m_s$.

### 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 $P_\gamma$ denote the predicate identifying a region $\gamma$.

**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 $\Gamma_{M_1}$ and $\Gamma_{M_2}$. The corresponding set of predicates is obtained by "anding" all possible pairs of predicates from both the structural components. For example, Figure 5 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 $\Gamma_{M_1}$ and $\Gamma_{M_2}$ 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.

Once again, we consider the example in Figure 5. $T_3$'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 $\Gamma_M$ be a common refinement of $\Gamma_{M_1}$ and $\Gamma_{M_2}$. 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. (For a counter example to see that the above lemma is not valid for $g = g_{max}$, see the full paper [18].)

## 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 [26, 30]. 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. $\gamma$ by taking its intersection with $\gamma$. The measure w.r.t. a dataset $D$ for each region $\gamma^i_M$ focussed w.r.t. $\gamma$ 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. $\gamma$ 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. $\gamma$.

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 $\gamma$ may not be less than that over $\gamma'$. 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` ($\sqcup$): The structural union of two sets of regions $\Gamma_1$ and $\Gamma_2$ is given by their GCR $\bigwedge(\Gamma_1, \Gamma_2)$.
2. `Structural Intersection` ($\sqcap$): The structural intersection of $\Gamma_1$ and $\Gamma_2$ is the set of regions $\Gamma$ such that each region in $\Gamma$ is a member of both $\Gamma_1$ and $\Gamma_2$. This is identical to the standard intersection operation on sets.

3. `Structural Difference` $\ominus$: The structural difference of $\Gamma_1$ and $\Gamma_2$ 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 $\Gamma$, two datasets $D_1, D_2$, and a deviation function $\delta_{(f,g)}$, the `rank` operator $\rho(\Gamma, \delta_{(f,g)}, D_1, D_2)$[4] 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 $\theta^{top}$, $\theta^{n}$, $\theta^{min}$, and $\theta^{-n}$ 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 5. 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:

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

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:

$$\rho(\theta^{10}(\rho(\mathcal{P}(\mathcal{I}_1) \cap (\Gamma_{L_1} \sqcup \Gamma_{L_2})), \delta) \cup \theta^{10}(\rho(\mathcal{P}(\mathcal{I}_2) \cap (\Gamma_{L_1} \sqcup \Gamma_{L_2}))), \delta)$$
$$\underline{\theta^{20}(\rho(\mathcal{P}(\mathcal{I}_1) \cup \mathcal{P}(\mathcal{I}_2)) \cap (\Gamma_{L_1} \sqcup \Gamma_{L_2}), \delta)}$$

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

### 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.

#### 5.2.1 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 $\Gamma_T$.

**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 [13] 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:

| 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 $s_{i+1}$

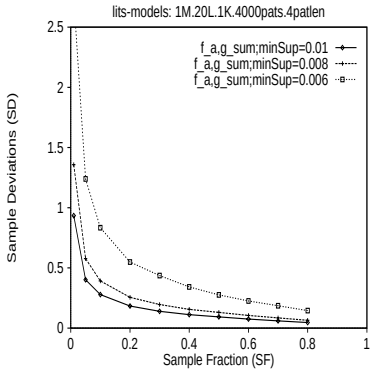



Figure 7: SD vs SF

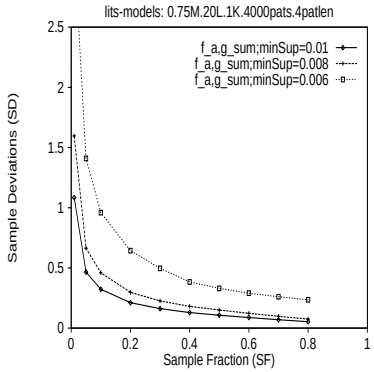



Figure 8: SD vs SF

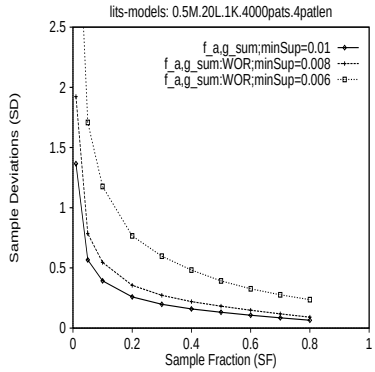



Figure 9: SD vs SF

(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. 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 3.4) 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. 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]. (We omit the details due to space constraints. See the full paper for details [18].)

### 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.

| 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 $s_{i+1}$

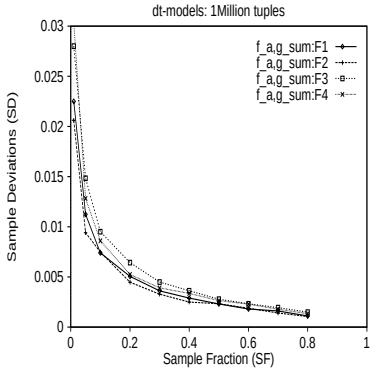



Figure 10: SD vs SF

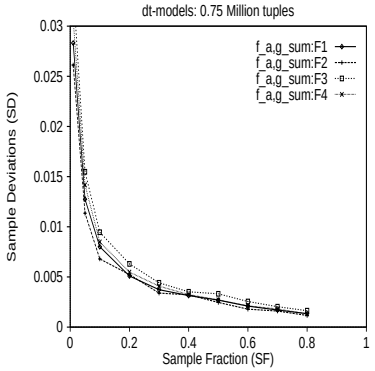



Figure 11: SD vs SF

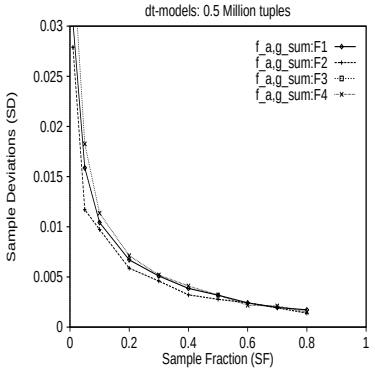



Figure 12: SD vs SF

#### 6.1.1 lits-models

We used the synthetic data generator from the IBM Quest Data Mining group[5]. We use $N$M.$t_l$L.$|\mathcal{I}|$I.$N_p$pats.$p$plen to refer to a dataset with $N$ million transactions, average transaction length $t_l$, $|\mathcal{I}|$ thousand items, $N_p$ thousand patterns, 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 the full paper [18].) 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 7, 8, and 9 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 7, 8, and 9 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%.

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

#### 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 $N$`M.F`$num$ 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 [20] 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 the 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 10, 11, and 12 show the plots for different classification functions (F1, F2, F3, and F4) in the IBM data generator and for varying 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-

| Dataset | $\delta$ | % $sig(\delta)$ | $\delta^*$ | Time for $\delta$ | Time for $\delta^*$ |
|---|---|---|---|---|---|
| $D_{(1)}$ | 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 13: Deviation with D: 1M.20L.1I.4pats.4plen

| ID | $\delta$ | % $sig(\delta)$ |
|---|---|---|
| $D_{(1)}$ | 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 14: Deviation with D: 1M.F1

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 $\delta^*$.

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 13 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 (row `(5)`).

The last three columns in Figure 13 show that $\delta^*$ delivers a good estimate instantaneously. The equality of 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 3.4) 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

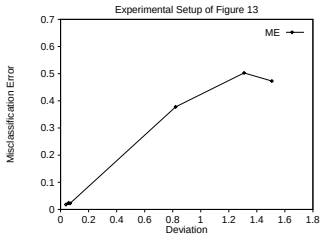



Figure 15: $Deviation$ vs. ME

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 15, we plot the misclassification error (ME) for the tree constructed from $D$ w.r.t. 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.

## 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 [15, 23, 29, 35, 38]. In contrast to the work on clustering, [6, 25, 27] concentrated primarily on discovering outliers in a dataset.

Interestingness measures to monitor variation in a single pattern were proposed in [36]. A similar problem of monitoring the support of an individual itemset was addressed in [4, 10]. 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 Transactions 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, Johannes Gehrke, Raghu Ramakrishnan, and Wei-Yin Loh. A framework for measuring changes in data characteristics. http://www.cs.wisc.edu/ vganti/pods99-full.ps, November 1998.

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

[20] 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.

[21] 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.

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

[23] 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.

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

[25] 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.

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

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

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

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

[37] 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.

[38] 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.