A general framework for estimating similarity of datasets and decision trees: exploring semantic similarity of decision trees

Irene Ntoutsi † Alexandros Kalousis * Yannis Theodoridis †

Abstract

Decision trees are among the most popular pattern types in data mining due to their intuitive representation. However, little attention has been given on the definition of measures of semantic similarity between decision trees. In this work, we present a general framework for similarity estimation that includes as special cases the estimation of semantic similarity between decision trees, as well as various forms of similarity estimation on classification datasets with respect to different probability distributions defined over the attribute-class space of the datasets. The similarity estimation is based on the partitions induced by the decision trees on the attribute space of the datasets. We use the proposed framework in order to estimate the semantic similarity of decision trees induced from different subsamples of classification datasets; we evaluate its performance with respect to the empirical semantic similarity, which we estimate on the basis of independent hold-out test sets. The availability of similarity measures on decision trees opens a wide range of possibilities for meta-analysis and meta-mining of the data mining results.

1 Introduction

Decision tree (DT) models are one of the most popular learning paradigms in the area of data mining thanks to a number of attractive properties they possess, such as scalability to large datasets and relative easiness of interpretation, provided that their size does not exceed certain limits. On the other hand, they are also notorious for their instability; small changes in the training dataset may result to completely different DTs that contain different tests on the predictive attributes or even different predictive attributes. These DTs, though structurally different, may describe the same concept, i.e., they may be semantically similar or even identical to each other; in fact, two DTs are expected to be semantically similar if they have been induced from datasets that come from the same generating distribution. Semantic similarity in the presence of structural differences might arise for a variety of reasons, such as superficially different tests on attributes which are in fact equivalent, different attributes that convey the same information due to attribute redundancy, or simply because the same concept can be described in different ways which are nevertheless semantically equivalent. To capture the degree of semantic similarity between DTs we need a measure of the semantic similarity of the concepts that they describe.

There is a plethora of reasons for which the definition of similarity measures between DTs and classification datasets is required. By far, the most important is on being able to report whether the differences observed in DTs induced from different training sets (which, however, are thought to come from the same data generating distribution) are only structural and do not correspond to semantic differences, or whether the concepts described by the DTs are indeed semantically different. In the latter case, a quantification of this semantic difference would be useful. Moreover, the availability of a similarity measure on classification models makes it possible to apply a number of standard mining tasks on classification models rather than on raw data, resulting in what we could call meta-analysis or meta-mining tasks. For example, the semantic similarity measures can be used to cluster different sites into groups of similar behavior according to the DT models learned locally on each of the sites, e.g. clustering the different branches of a bank according to the credit strategy they adopt.

Similarity could be also employed in order to study the effect of the DT learning parameters, like pruning level, on the resulting models or to compare a DT model to a golden standard model. Also, in case of dynamic data, like data streams, similarity could be employed in order to monitor the evolution of the induced models or classification datasets across the time axis. A crucial question in this case is whether the concept, which is captured through the induced models, remains (about) the same or there are concept drifts in the population.

In this paper, we propose a general similarity estimation framework that includes as special cases i) the estimation of semantic similarity between DTs and

*Computer Science Department, University of Geneva, Switzerland

†Department of Informatics, University of Piraeus, Greece

ii) the estimation of similarity between different probability distributions that govern different classification datasets, namely the marginal distribution of the attributes, the joint attributes-class probability distribution and the attributes-conditional class distribution. The framework is based on the comparison of the partitionings that DTs define over a given attribute space considering the probability distribution of the data space over that partitioning. Similar ideas have been previously used for dataset comparison, however, to the best of our knowledge this is the first time that the semantic similarity of DTs is explored. Depending on the available information regarding the probability distribution that generated the data, we get different instantiations of the DT semantic similarity measure. To evaluate the proposed measures we compare them with the empirical semantic similarity, which is estimated by applying the DTs on independent test sets.

The rest of the paper is organized as follows: In Section 2 we present the related work and some preliminaries on DTs . The proposed similarity framework is described in Section 3. The experimental evaluation of the different instantiations for the DT similarity measures is reported in Section 4. Finally, Section 5 discusses conclusions and outlook.

2 Preliminaries on decision trees and related work

Consider a classification problem described through a vector of predictive attributes $A = (a_1, a_2, ..., a_m)$ and a class attribute C. Each predictive attribute, a_i, has a domain, $d(a_i)$ and the domain of the class attribute is $d(C) = \{c_1, c_2, ..., c_k\}$, where k is the number of classes.

To built a decision tree, a set D of training examples is provided as input to the DT induction algorithm. Training examples are drawn from the joint distribution $P(A, C)$ of the predictive attributes and the class attribute. The Cartesian products $S_A = d(a_1) \times d(a_2) \ldots \times d(a_m)$ and $S_{(A,C)} = S_A \times d(C)$ define the *attribute* and *attribute-class* spaces, respectively. Training examples have thus the form $(\mathbf{x}, y) \in S_{(A,C)}$, where $\mathbf{x} \in S_A$ and $y \in d(C)$. Let $U(A)$ denote the uniform distribution over the attribute space and $P(A) = \sum_C P(A, C)$ denote the marginal distribution also defined over the attribute space.

Although DT induction is an extensively studied research area, limited work has been done on the problem of DT comparison and more precisely on the computation of semantic similarity between DTs; the only notable exception is [7]. More recently, several approaches have been proposed that utilize DT comparison as a means for dataset comparison, e.g. [3, 5].

Turney [7] presented a framework for evaluating the *stability* of a classification algorithm, namely the degree to which it generates repeatable results, when trained on different datasets drawn from the joint distribution $P(A, C)$. To quantify stability, Turney uses a semantic similarity measure called *agreement*. The agreement of two classifiers is defined as the probability of producing the same prediction over instances drawn from $U(A)$. Note that, according to Turney, agreement is measured over instances drawn from $U(A)$ and not from $P(A, C)$; the underlying reason is that the agreement should be examined over all possible input worlds. Turney estimates the agreement of DTs empirically, by applying them on artificial test sets of instances drawn from the $U(A)$ distribution.

Recently, several change detection methods have been proposed that utilize DTs for *dataset comparison*. The intuition behind these approaches is that the DT models capture interesting characteristics of the datasets and thus they can be used for similarity assessment between datasets. All the methods in this category follow the same rationale: they combine DTs to induce a "finer" DT structure, and then they compare the distributions of the two datasets over this (common) "finer" structure. Below, we describe some representative approaches in this category.

Ganti et al. [3] propose the FOCUS framework for measuring the deviation between two datasets D_1, D_2 in terms of the corresponding decision trees DT_1, DT_2. Each DT defines, via its leaf nodes, a set of non-overlapping regions over the attribute space, whereas by overlaying the regions of the two DTs a "finer" structure arises. The authors compute the probability of each region in the overlay by querying the original raw datasets. Then, the distance between the two datasets is computed by aggregating, for each region in the overlay, the difference in the region probability estimations between the two datasets.

Wang and Pei [9] quantify changes between two datasets with class labels using as a common structure for the comparison a set of *random histograms*. The instances of the two datasets are projected into this structure and changes in their distributions are detected.

Recently, Pekerskaya et al. [5] proposed a method for mining changing regions between two datasets. A region is characterized as changing if it appears under different class labels in the two datasets. The authors extend the traditional DT structure by further splitting the leaf nodes through clustering. The resulting model is called *cluster-embedded DT* and provides a better approximation of the attribute space probability distribution comparing to the approximation of a (simple) DT. After extracting the cluster-embedded DT structure for each dataset, the overlay of the two structures

is computed and its statistics are estimated without requerying the original raw datasets, as in FOCUS [3]. Rather, the authors approximate the measure component of each region in the overlay by employing the statistics of the corresponding cluster-embedded DTs.

Finally, there is a considerable amount of work on comparing tree structures based on the *edit distance*, e.g. [10]. These approaches are based on counting the number and the cost of edit operations (insert, delete, update) that are required in order to convert one tree into the other. They work with symbolic trees where the nodes are labeled with symbols from a given alphabet. In DTs, though, the nodes are more complex since they include conditions over the symbols-attributes and furthermore, each DT path is assigned a weight based on the number of instances that follow that path.

3 The similarity estimation framework

A decision tree DT induced from a dataset D partitions the attribute space into a set of non-overlapping regions $R_{DT} = \{r_i, i = 1 \ldots |R_{DT}|\}$, via its leaf nodes. The partition R_{DT} can be considered as an approximation of the joint attribute-class probability distribution in the form of a histogram (Section 3.1). Each bin of the histogram corresponds to a region of the partition and respectively, to a leaf node of the decision tree. A bin-region is defined by the tests on the predictive attributes encountered on the path from the root to the leaf node associated with that region. The frequencies of a given bin are the class counts of the instances that belong to the given bin. Different decision trees result in different partitions; in Section 3.2 we show how to derive the overlay partition of two decision trees and how to estimate its statistics depending on whether we have access to the original raw datasets or not. Based on the overlay partition, we define various similarity measures for decision trees and classification datasets (Section 3.3).

3.1 Decision tree partitions

Each region $r \in R_{DT}$ is characterized by a structure and a measure component that are directly derived from the decision tree.

The *structure component* of the region is defined as the conjunction of the test conditions on the attributes along the corresponding tree path from the root to the leaf node associated with that region:

$$r.s := \{\wedge t(a_i), i = 1 \ldots m\}$$

Test conditions are usually numeric and can be expressed in the form $t(a) := \min_a(r) \leq a \leq \max_a(r)$ denoting the min and max values of attribute a in region r. Let us also define the length of a test condition on a as: $|t(a)| := \max_a(r) - \min_a(r)$ and the length of the domain of a as: $|dom(a)| := \max_a - \min_a$. Note here that, if an attribute a is not included in the structure component $r.s$ of a leaf node, i.e., no test on that attribute has been included in the path from the root to the leaf node during the training phase, then the test condition on that attribute is $t(a) := \min_a \leq a \leq \max_a$, i.e., a can take any value from its domain. Thus, the structure component of a region contains test conditions over all (i.e., m) predictive attributes of the problem.

The *measure component* of a region is defined as the number of training instances that fall into this region for each class, and it depends on the training set D:

$$\mathbf{r.m_D} := [n_{c_1}, n_{c_2}, \ldots, n_{c_k}] \tag{3.1}$$

where $n_{c_i}, i = 1 \ldots k$ is the number of instances that fall into region r and belong to class c_i. The size of the measure component is: $|r.m_D| = \sum_{1 \leq i \leq k} n_{c_i}$ and the class $r.cl$ assigned to the region r is given by: $r.cl = \arg\max_{c_i} r.m_D$.

The *probability of a region* represents the probability that some instance of the problem will follow the corresponding DT path. Formally, this probability is given by: $P(r) = \int_r P(A)dA$, where $P(A)$ is the probability density function of the instances. However, since we do not have access to the exact form of $P(A)$, we should use the data to estimate it. More specifically, if we consider the dataset D used for the construction of the DT, we can make a dataset dependent estimation of $P(r)$ as follows [1] :

$$\mathbf{P_D(r)} = \frac{|\mathbf{r.m_D}|}{N_D} \tag{3.2}$$

This estimation is simply the percentage of the training set instances that fall in region r. The vector:

$$\mathbf{P_D(A)} = [\mathbf{P_D(r_i)} | r_i \in R_{DT}] \tag{3.3}$$

is an approximation of $P(A)$, from the dataset D.

Except for $P(A)$, we can also approximate $P(A, C)$ by exploiting the measure component of the regions, which describe the distribution of training set instances within the different problem classes. The matrix:

$$\mathbf{P_D(A, C)} = [\frac{\mathbf{r_i.m_D}}{N_D} | r_i \in R_{DT}] \tag{3.4}$$

in which each row corresponds to a region $r_i \in R_{DT}$ and each column to a class $c_j \in C$, is an approximation of the joint distribution, $P(A, C)$ from the dataset D.

[1] We denote the actual distribution by P and its estimation by $\mathbf{P}$.

Furthermore, the measure component can provide us with an estimation of the conditional probability of the classes given the region r:

$$\mathbf{P_D}(\mathbf{C}|\mathbf{r_i}) = \frac{\mathbf{r_i.m_D}}{|\mathbf{r_i.m_D}|} \tag{3.5}$$

Then, the estimate of the attributes conditional distribution of the class is the matrix:

$$\mathbf{P_D}(\mathbf{C}|\mathbf{A}) = [\mathbf{P_D}(\mathbf{C}|\mathbf{r_i})|r_i \in R_{DT}] \tag{3.6}$$

where each cell of the matrix corresponds to the probability of observing a specific region under a specific class.

3.2 Decision tree partitions overlay Let R_{DT_1} and R_{DT_2} be the partitions defined by the decision trees DT_1 and DT_2, respectively. Overlaying the two partitions, a finer partition $R_{DT_1 \times DT_2}$ arises, where each region r in it is the result of overlaying some region $r_i \in R_{DT_1}$ with some region $r_j \in R_{DT_2}$, that is $r = r_i \cap r_j$. The goal is to estimate the region probability $P(r)$ and the region-class probability $P(r, c)$ for each region $r \in R_{DT_1 \times DT_2}$ and each class $c \in C$. To this end, we rely on the observation that each region r in the overlay is also a hyperectangle and thus it can be described through a structure and a measure component.

3.2.1 Structure component of the overlay regions The *structure component* of the overlay region $r_i \cap r_j$ is easily defined through the intersections of the DT regions that participate in its formation:

$$r_i \cap r_j.s := \{\wedge t(a_i), i = 1 \dots m\}$$

$$t(a) := \min_a(r_i \cap r_j) \leq a \leq \max_a(r_i \cap r_j)$$

$$\min_a(r_i \cap r_j) := \max(\min_a(r_i), \min_a(r_j))$$

$$\max_a(r_i \cap r_j) := \min(\max_a(r_i), \max_a(r_j))$$

If $\max_a(r_i \cap r_j) \leq \min_a(r_i \cap r_j)$, the overlay region $r_i \cap r_j$ is not defined since the regions are disjoint.

3.2.2 Measure component of the overlay regions The estimation of the measure component of $r_i \cap r_j$ is dataset dependent; the obvious choices for the dataset are D_1, D_2 and $D_1 \cup D_2$. However, even if we do not have anymore access to any of these datasets, we can still estimate the measure component of the overlay regions based on the measure components of the regions of the original partitions R_{DT_1} and R_{DT_2}.

Data dependent probability estimation: If we have access to the original raw datasets, we can get the *exact* measure component of the overlay regions by simply projecting each dataset $D \in \{D_1, D_2, D_1 \cup D_2\}$ on $R_{DT_1 \times DT_2}$. That is:

$$\begin{aligned} \mathbf{r_i} \cap \mathbf{r_j.m_D} &= [n'_{c_1}, \dots, n'_{c_k}], \\ \textit{where } n'_{c_i} &= |\{(\mathbf{x}, c_i)\}|, \\ \mathbf{x} &\in r_i \cap r_j, \mathbf{x} \in D \end{aligned} \tag{3.7}$$

which simply gives us the number of training instances that fall within the $r_i \cap r_j$ region for the D dataset for each of the problem classes.

Pattern dependent probability estimation: Even if we do not have access to the original raw datasets, we can still make an estimation of the expected measure for each region $r_i \cap r_j \in R_{DT_1 \times DT_2}$ using the measure components of the original regions $r_i \in R_{DT_1}$ and $r_j \in R_{DT_2}$ (which are derived directly from the DTs). The expected measure of $r_i \cap r_j$ according to D_1 is:

$$\mathbf{r_i} \cap \mathbf{r_j.m_{D_1}} = \mathbf{r_i.m_{D_1}} \frac{V(r_i \cap r_j)}{V(r_i)} \tag{3.8}$$

where the term $\frac{V(r_i \cap r_j)}{V(r_i)}$ represents the relative volume of the intersection region $r_i \cap r_j$ with respect to the volume of the region r_i. Since the regions established by a DT are axis parallel hyper-rectangles it holds that:

$$V(r) = \prod_{a_i} \frac{|t(a_i)|}{|dom(a_i)|}$$

where the term $\frac{|t(a_i)|}{|dom(a_i)|}$ represents the relative importance of attribute a_i in region r. If we assume a uniform distribution $U(A)$ of the instances over the attribute space then $V(r) = P(r)$. In Equation 3.8, though, we adopt an intermediate assumption, namely that the D_1 instances are uniformly distributed within the region r_i of R_{DT_1}, instead of being uniformly distributed within the whole attribute space. As in Equation 3.8, the expected measure of $r_i \cap r_j$ according to D_2 is:

$$\mathbf{r_i} \cap \mathbf{r_j.m_{D_2}} = \mathbf{r_j.m_{D_2}} \frac{V(r_i \cap r_j)}{V(r_j)} \tag{3.9}$$

Finally, if we assume that the two datasets come from the same distribution $P(A)$, we can get the expected measure of $r_i \cap r_j$ according to the union, $D_1 \cup D_2$:

$$\mathbf{r_i} \cap \mathbf{r_j.m_{D_1 \cup D_2}} = \mathbf{r_i} \cap \mathbf{r_j.m_{D_1}} + \mathbf{r_i} \cap \mathbf{r_j.m_{D_2}}$$

So far, we have shown how we can estimate the probabilities of the overlay regions depending on whether we have access to the original raw datasets or not. As with the single DT partition case (c.f. Section 3.1), we can use these estimations to approximate

the distributions $P(A), P(A,C), P(C|A)$. Depending on which dataset, $D \in \{D_1, D_2, D_1 \cup D_2\}$, we use to calculate the measures of the overlay regions, we get the corresponding estimations of $\mathbf{P_D}(\mathbf{A}), \mathbf{P_D}(\mathbf{A},\mathbf{C})$ and $\mathbf{P_D}(\mathbf{C}|\mathbf{A})$ under the $R_{DT_1 \times DT_2}$ partition. To distinguish between the case where the measure components are computed by accessing the original raw datasets (Equation 3.7) or under the uniform region distribution assumption (Equations 3.8, 3.9), we use the superscripts ^{Q} and ^{U} respectively.

3.3 Similarity measures on decision trees and datasets In the previous section we described methods for the estimation of $\mathbf{P_D}(\mathbf{A}), \mathbf{P_D}(\mathbf{A},\mathbf{C})$ and $\mathbf{P_D}(\mathbf{C}|\mathbf{A})$ under the $R_{DT_1 \times DT_2}$ partition and for the different datasets $D \in \{D_1, D_2, D_1 \cup D_2\}$. These estimations can be used to compute similarities between either DTs or datasets.

Before we proceed with the definition of the actual similarity measures, we first provide a similarity function between histograms, since all our estimations come in the form of histograms. Let P, Q be the probability density estimations for a random variable X from two different populations, in the form of histograms. We assume that P and Q are defined over the same bins. The *affinity coefficient* between P and Q is given by:

$$s(P,Q) = \sum_i \sqrt{P_i Q_i}$$

Based on the affinity coefficient and the different overlay partition statistics, we can now define a number of similarity measures between DTs and datasets:

Case a: We can measure the *similarity of two datasets* D_1, D_2 with respect to their attribute space probability distributions $P_{D_1}(A)$, $P_{D_2}(A)$ by directly computing their affinity coefficient:

$$s(\mathbf{P_{D_1}}(\mathbf{A}), \mathbf{P_{D_2}}(\mathbf{A})) \tag{3.10}$$

This similarity measure can be used to determine if the two datasets were generated from the same distribution $P(A)$. The estimations $\mathbf{P_{D_i}}(\mathbf{A}), i = \{1,2\}$ can be either $\mathbf{P^Q_{D_i}}(\mathbf{A})$ or $\mathbf{P^U_{D_i}}(\mathbf{A})$ depending on whether raw data access is allowed or not.

Case b: We can measure the *similarity of two DTs* DT_1, DT_2 with respect to their predictions. This is a measure of their *semantic similarity*, i.e., how similar are the concepts described by the DTs, and corresponds to the percentage of times that they produce the same predictions on instances drawn from a given attribute space distribution.

We first define the vector:

$$\mathbf{I}(\mathbf{C}|\mathbf{A}) = [I(r_i.cl, r_j.cl) | r_i \cap r_j \in R_{DT_1 \times DT_2}]$$

which indicates whether the two DTs agree or disagree in their predictions over the regions of the overlay partition $R_{DT_1 \times DT_2}$. $I(r_i.cl, r_j.cl)$ returns 1 if the predictions of the two DTs regarding the region $r_i \cap r_j$ are the same, i.e., $r_i.cl = r_j.cl$, otherwise it returns 0. The inner product [2]:

$$S(DT_1, DT_2) = \mathbf{I}(\mathbf{C}|\mathbf{A})' \mathbf{P}(\mathbf{A}) \tag{3.11}$$

computes the similarity in the predictions of DT_1, DT_2 under the $P(A)$ distribution. The similarity score equals to the sum of probabilities of the $r_i \cap r_j$ regions for which the trees agree in their predictions.

One issue that rises here is which estimation of $P(A)$ we should employ. Possible choices include:

- *the uniform distribution,* $U(A)$. In this case the agreement will be examined over all possible input worlds. Under this assumption, the probability of a region $r_i \cup r_j$ is given by its hyper-volume. Thus, the similarity between two DTs equals to the total volume of the regions in which the two DTs agree in their predictions. In this case, Equation 3.11 gives, in a closed form, the semantic similarity between the two DTs as it was defined by Turney [7]. Note however that, in contrast to [7], we *do not require* for this estimation the generation of an artificial test set drawn from $U(A)$.
- *a dataset dependent distribution* $\mathbf{P_D}(\mathbf{A})$, where D can be one of the D_1, D_2 and $D_1 \cup D2$ datasets. In this case, instances are assumed to follow the distribution of the dataset $D \in \{D_1, D_2, D_1 \cup D_2\}$. The union, $D_1 \cup D2$, is the most appropriate choice if the trees are generated from datasets following the same distribution and we are interested in evaluating their similarity under that distribution.
- finally, $P(A)$ might be a distribution that is *different* from the distributions that govern the training sets.

Case c: We can also measure the similarity of two datasets with respect to the attribute conditional probability distribution of the class attribute $P(C|A)$ that the DTs, which were induced from these datasets, impose over the attribute space. We first define the vector:

$$\mathbf{S}(\mathbf{C}|\mathbf{A}) = [s(\mathbf{P_{D_1}}(\mathbf{C}|\mathbf{A})[r_i,], \mathbf{P_{D_2}}(\mathbf{C}|\mathbf{A})[r_j,]) | r_i \cap r_j \in R_{DT_1 \times DT_2}]$$

[2] We denote by X' the inverse of matrix X.

$\mathbf{S}(\mathbf{C}|\mathbf{A})$ has the same structure as $\mathbf{I}(\mathbf{C}|\mathbf{A})$, but the 0/1 similarity function $I(.,.)$ has been replaced by $s(.,.)$, which computes the similarity of the attribute conditional class distributions of the $r_i \cap r_j$ region in the D_1 and D_2 datasets. The inner product:

$$S(D_1, D_2) = \mathbf{S}(\mathbf{C}|\mathbf{A})'\mathbf{P}(\mathbf{A}) \tag{3.12}$$

provides a measure of the similarity of the two datasets with respect to their attribute conditional class distributions under an attribute space that follows the $P(A)$ distribution.

Note here that this measure is similar to the measure that is used in [5] to rank the changing regions between two datasets. In fact, their approach is equivalent to introducing a distance measure of the form:

$$D(D_1, D_2) = \mathbf{D}(\mathbf{C}|\mathbf{A})'\mathbf{P}(\mathbf{A})$$

where $D(C|A)$ has the same structure as $S(C|A)$ but the similarity function is replaced by the Euclidean distance and $P(A)$ is approximated by:

$$\mathbf{P}(\mathbf{A}) = \frac{1}{2}(\mathbf{P_{D_1}}(\mathbf{A}) + \mathbf{P_{D_2}}(\mathbf{A}))$$

However, the authors in [5] do not go as far as to define the $D(D_1, D_2)$. They rather define the product of $D(C|A)$ and $P(A)$, i.e., the vector consisting of the pairwise products of the coordinates of the two vectors, and use that in order to rank regions according to their level of change from one dataset to the other.

Case d: Finally, we can measure the similarity of the joint attribute-class probability distribution of the two datasets $P_{D_1}(A, C)$, $P_{D_2}(A, C)$ by simply applying the affinity coefficient:

$$s(\mathbf{P_{D_1}}(\mathbf{A}, \mathbf{C}), \mathbf{P_{D_2}}(\mathbf{A}, \mathbf{C})) \tag{3.13}$$

$\mathbf{P_{D_i}}(\mathbf{A}, \mathbf{C})$ is the estimation of $P_{D_i}(A, C)$ under the overlay partition. Note here that if the two datasets came from the same $P(A)$ distribution then it can be easily shown that this measure is equivalent to $S(D_1, D_2)$ given in Equation 3.12. In fact this is the approach that was followed by FOCUS [3] for measuring dataset deviation. The difference lies in the fact that, instead of the affinity coefficient, FOCUS employs a difference function f (e.g. absolute or relative difference) to compute the measure similarity within each region and an aggregation function g (e.g. sum or max) to aggregate the scores of the overlay regions into an overall score.

In this section, we presented a general framework for similarity estimation between either DTs or datasets. Under this framework, we can estimate the similarities of classification datasets with respect to a number of probability distributions: i) the attribute space distribution $P(A)$ (Equation 3.10), ii) the class attribute conditional distribution $P(C|A)$ (Equation 3.12) and iii) the joint attribute-class distributions $P(A, C)$ (Equation 3.13). We can also use this framework in order to estimate the semantic similarity of DTs (Equation 3.11) under different assumptions for the attribute space probability distribution. It is this direction that we are going to explore and evaluate in more detail in the next section.

4 Evaluation of the proposed similarity measure on decision trees

The semantic similarity of any two classification models M_1, M_2 is defined as the fraction of times that the two models produce the same predictions over instances generated from a given attribute space probability distribution $P(A)$. As already mentioned, Turney [7] defined a semantic similarity measure for classification models, called agreement, as the probability that they will produce the same predictions over all possible instances drawn from the uniform distribution on the attribute space, $U(A)$. Turney estimates the agreement between two classification models empirically, by applying both of them on a test set D_H of instances drawn from the $U(A)$ distribution, and computing the percentage of times that they produce the same predictions. The argument for employing $U(A)$, instead of the distribution $P(A)$ that generated the data, was that the agreement of two concepts should be examined in all possible input worlds. Contrary, we argue that in a real world application what is more important is not the similarity of the DTs in all possible worlds, but rather similarity in the world in which the data exist. So, unlike [7], in order to estimate the semantic similarity, we draw the D_H dataset from $P(A)$, the distribution that governs the attribute space. We denote by $S_H(DT_1, DT_2)$ the semantic similarity between DT_1 and DT_2; this similarity is empirically estimated on the D_H dataset by applying the two DTs on D_H and computing the number of times that they produce the same predictions. $S_H(DT_1, DT_2)$ provides the ground truth to which we will compare the proposed DT semantic similarity measures.

4.1 Datasets We experimented with six different datasets, a short description of which is given in Table 1. The different *mfeat* datasets are versions of the same pattern recognition problem in which the goal is to classify handwritten numerals. The versions correspond to different features used to describe the numerals: in *mfeat-factors*, attributes are profile correlations,

dataset	# inst	# attrs	# classes
mfeat-factors	2,000	21	10
mfeat-fourier	2,000	76	10
mfeat-karhunen	2,000	64	10
mfeat-zernike	2,000	47	10
segment-challenge	2310	19	7
waveform-5000	5,000	40	3

Table 1: Description of datasets.

in *mfeat-zernike* zernike moments, in *mfeat-karhunen* Karnhunen-Love coefficients and in *mfeat-fourier* fourier coefficients of the character shapes [4]. *Waveform-5000* is an artificial dataset where classes correspond to different types of waves [2]. In the *segment-challenge* dataset, [8], features are high level descriptors of regions of images and the goal is to classify each region to the correct class, e.g. sky, grass.

4.2 Experimental setup We need a systematic way to generate DTs that exhibit varying degrees of semantic similarity. To this end, we randomly divide a given dataset D in two parts, a training set D_T used during the model construction phase, and a test set D_H used as the hold out set for the computation of S_H ($|D_H| = \frac{1}{3}|D|$). Then, we create random sub-samples of the D_T of size p ($p = 5\% \ldots 95\%$) with a step of 5%. On each sub-sample DT_p, a decision tree is trained and compared to the DT that was created on the complete training set, DT_{100}. Then, we compute the semantic similarity between the complete DT and the sampled one, i.e., $S_H(DT_p, DT_{100})$, on the hold out set D_H.

First of all, we should verify that the procedure we employed for the generation of the different DTs DT_p indeed results in trees that exhibit varying levels of semantic similarity with respect to DT_{100}. We expect $S_H(DT_p, DT_{100})$ to increase as p increases and approaches 100%, since the training set D_p used in the construction of DT_p becomes more and more similar to the training set D_{100} used in the construction of DT_{100}. This is indeed the case as one can see in Figure 1, where we plot S_H as a function of the sampling size p; there is a smooth increase in the values of S_H as p increases towards 100%.

4.3 Evaluating semantic similarity The goal of the experimental evaluation that we present in this section is to examine how the different semantic similarity measures that we propose correlate with S_H.

The DT semantic similarity measure $S(DT_1, DT_2)$ that we propose (Equation 3.11) depends on the estimation of the $P(A)$ distribution that governs the attribute. In fact, the computation of similarity makes sense for a

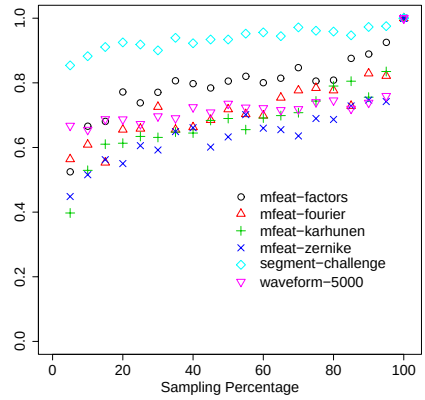

Figure 1: Evolution of $S_H(DT_p, DT_{100})$

given world, in which a specific distribution $P(A)$ holds for the attribute space. Then, $S(DT_1, DT_2)$ is simply the sum of the probability densities, under the chosen $P(A)$, of the $r_i \cap r_j$ regions in which the two DTs agree. As already mentioned, under the uniform distribution assumption this sum equals to the sum of the hypervolumes of these regions. Moreover, under that assumption, $S(DT_1, DT_2)$ provides the semantic similarity of Turney [7] *without* having to apply the learned models on the hold out set. We will not further examine the uniform assumption as a possible estimation for $P(A)$. Instead, we will experiment with three different instantiations of $S(DT_1, DT2)$ that differ with respect to the estimation of $P(A)$ they employ. In particular, we will investigate the following estimations for $P(A)$:

- $P^U_{D_1 \cup D_2}$: this is the estimation of $P(A)$ that we get when the measure components are computed under the uniform region distribution assumption, as in Equations 3.8, 3.9.
- $P^Q_{D_1 \cup D_2}$: this is the estimation of $P(A)$ that we get when the measure components are computed from the direct application of the overlay partition $R_{DT_p \times DT_{100}}$ on the D_p and D_{100} datasets.
- P^Q_H: this is the estimation of $P(A)$ that we get from the direct application of the overlay partition $R_{DT_p \times DT_{100}}$ on the hold out set D_H.

Each of these estimations, P^Y_X, of P(A) results in a different instantiation of $S(DT_1, DT_2)$ which we denote by $S_{P^Y_X}(DT_1, DT_2)$. We should note that the order in which the different P^Y_X are listed reflects an increasing amount of knowledge about the $P(A)$ distribution that

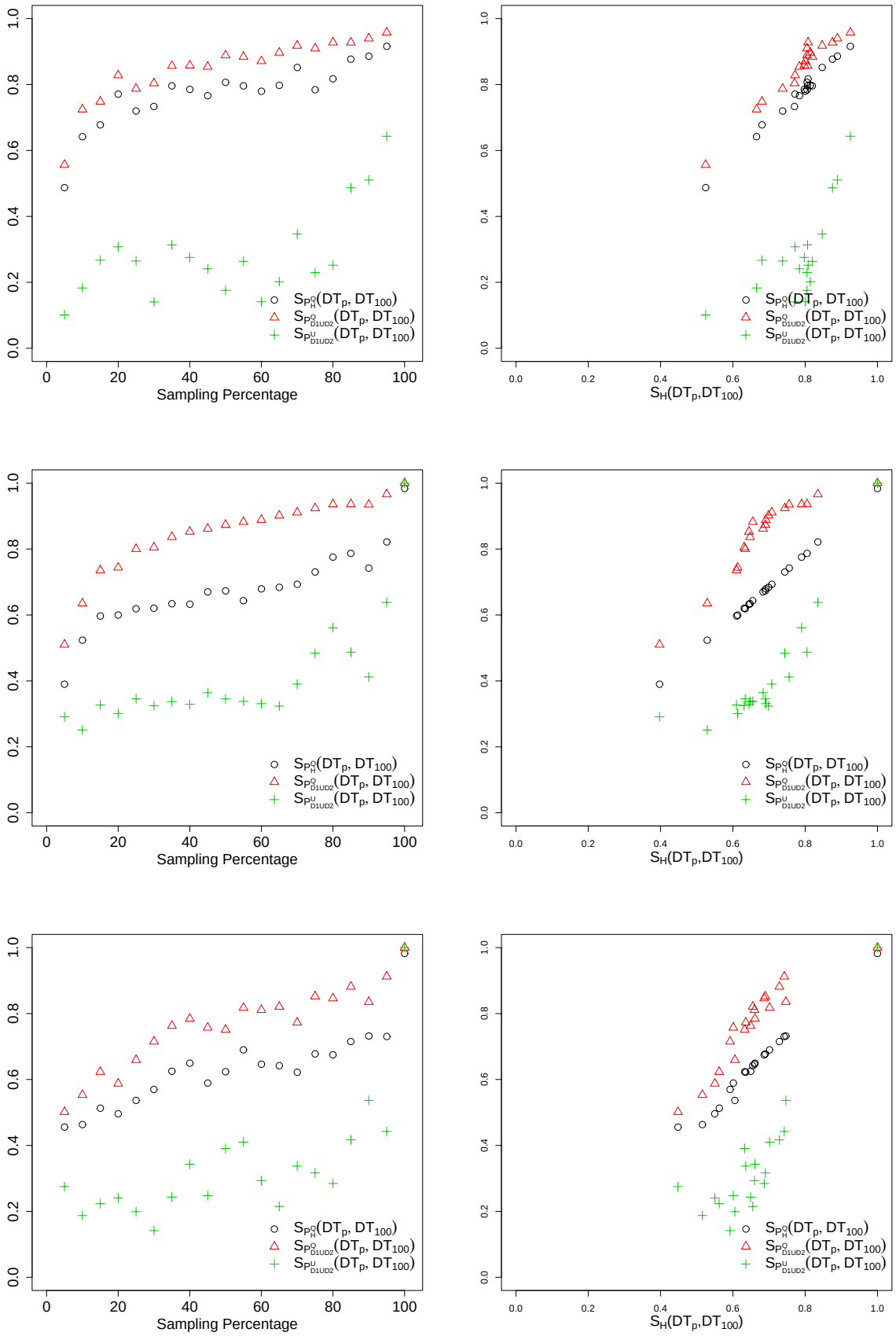

Figure 2: Evolution of the DTs semantic similarity measures with the sampling rate (first column) and with S_H (second column) for datasets: mfeat-factors (top), mfeat-karhunen (middle), mfeat-zernike (bottom)

governs the computation of the semantic similarity S_H, which we use in order to evaluate the proposed similarity measures. $P^U_{D_1 \cup D_2}$ assumes the least knowledge about $P(A)$; to estimate the measure components of the overlay tree, it only relies on the analysis of the structures of the respective DTs, under the assumption of a uniform within region distribution. $P^Q_{D_1 \cup D_2}$ requires *querying* D_1 and D_2 in order to estimate the measure components of the overlay tree; as a result, its estimation of $P(A)$ is more precise than the one provided by $P^U_{D_1 \cup D_2}$. Finally, P^Q_H has complete knowledge of $P(A)$, as this knowledge underlies in the D_H dataset, since we derive it by *querying* D_H. As a result, $S_{P^Q_H}(DT_1, DT_2)$ is expected to correlate perfectly with S_H. In that sense, $S_{P^Q_H}$ represents the ideal behavior that we get when we have knowledge of the true $P(A)$.

For each $S_{P^Y_X}(DT_p, DT_{100})$, we show how its value varies with respect to the sample size p, in the first column of Figures 2, 3. All the measures exhibit a similar pattern; similarity increases as p increases. More particular, $S_{P^Q_{D_1 \cup D_2}}$ and $S_{P^Q_H}$ have a very regular behavior, with an almost steady increase of values and small fluctuations. In case of the $S_{P^U_{D_1 \cup D_2}}$ similarity, the trend is also increasing but here the fluctuations can be considerably larger, as it happens in the *mfeat-zernike*, *mfeat-factors*, *segment-challenge*, *mfeat-fourier* datasets. $S_{P^Q_{D_1 \cup D_2}}$ is constantly overestimating DT similarity compared to $S_{P^Q_H}$, while $S_{P^U_{D_1 \cup D_2}}$ considerably underestimates it; recall here that $S_{P^Q_H}$ reflects the ideal behavior.

In the second column of Figures 2, 3, we see how the three different versions of $S_{P^Y_X}(DT_p, DT_{100})$ correlate with the actual evaluation measure $S_H(DT_p, DT_{100})$. As it was expected, $S_{P^Q_H}$ correlates perfectly since its estimation of $P(A)$ is taken from the D_H dataset on which $S_H(DT_p, DT_{100})$ is computed. Consequently, $S_{P^Q_{D_1 \cup D_2}}$ is constantly overestimating $S_H(DT_p, DT_{100})$, while $S_{P^U_{D_1 \cup D_2}}$ is considerably underestimating it. The performance of $S_{P^Q_{D_1 \cup D_2}}$ is quite close to the ideal performance of $S_{P^Q_H}$ with the most notable cases being *segment-challenge* and *mfeat-factors*, while the highest discrepancy appears in the case of *mfeat-karhunen*. Note here that datasets D_p, D_T and D_H are all drawn from the same $P(A)$ distribution. The discrepancy between the behavior of $S_{P^Q_{D_1 \cup D_2}}$ and $S_{P^Q_H}$ can be explained by the inaccuracy in the sampling procedure. As the number of instances increases, the behaviors of $S_{P^Q_{D_1 \cup D_2}}$ and $S_{P^Q_H}$ will converge since the estimations of $P(A)$ that the two methods employ will also converge. Alternatively, if we use repeated sampling over the D_H, D_T and D_p datasets and subsequently average over the different samples, the two measures would also converge. On the other hand, the behavior of $S_{P^U_{D_1 \cup D_2}}$ will be similar to that of $S_{P^Q_H}$ only to the level that the assumption of a within region uniform distribution is a valid assumption for the $P(A)$ governing D_H; nevertheless, as it is apparent for the datasets we have considered here, this is far from being a valid assumption.

4.3.1 Quantitative analysis of the measures

In order to quantify the behavior of each of the $S_{P^Y_X}(DT_p, DT_{100})$ we computed their Pearson correlation coefficient with $S_H(DT_p, DT_{100})$. The results are depicted in Table 2, where it seems that $S_{P^Q_{D_1 \cup D_2}}$ exhibits a very strong correlation with $S_H(DT_p, DT_{100})$. For most of the datasets, the correlation is higher than 0.9, with the notable exception of *waveform-5000* for which a low correlation coefficient is recorded. $S_{P^U_{D_1 \cup D_2}}$ has also a strong correlation with $S_H(DT_p, DT_{100})$ although not as strong as $S_{P^Q_{D_1 \cup D_2}}$, again with the remarkable exception of *waveform-5000* for which it exhibits its highest correlation value.

The Pearson correlation coefficient is an estimate of the linear correlation of two values, nevertheless it does not indicate how good predictor one variable is for the other. This is especially true in our case, since the pattern of linear correlation of any given $S_{P^Y_X}(DT_p, DT_{100})$ with $S_H(DT_p, DT_{100})$ changes from dataset to dataset as it is obvious from Figures 2, 3. In order to estimate the predictive value of the various $S_{P^Y_X}(DT_p, DT_{100})$ with respect to $S_H(DT_p, DT_{100})$, we compute their Mean Absolute Deviation (MAD). The MAD of two variables a and b for which we have N paired observations is given by:

$$MAD(a, b) = \sum_{i}^{N} \frac{|a_i - b_i|}{N},$$

The MAD results are given in Table 3. These results indicate the good predictive performance of $S_{P^Q_{D_1 \cup D_2}}$, its average error (MAD) in predicting $S_H(DT_p, DT_{100})$ is 0.1. The performance of $S_{P^U_{D_1 \cup D_2}}$ is considerably worse, its average MAD is roughly 0.3.

The goal of the current section was to compare and evaluate a number of different instantiations of a DT semantic similarity measure. The different instantiations are the result of different assumptions or different ways of estimating the attribute space distribution under which the semantic similarity computation will take place. In fact, the semantic similarity computation of two DTs makes sense if we can assume a specific

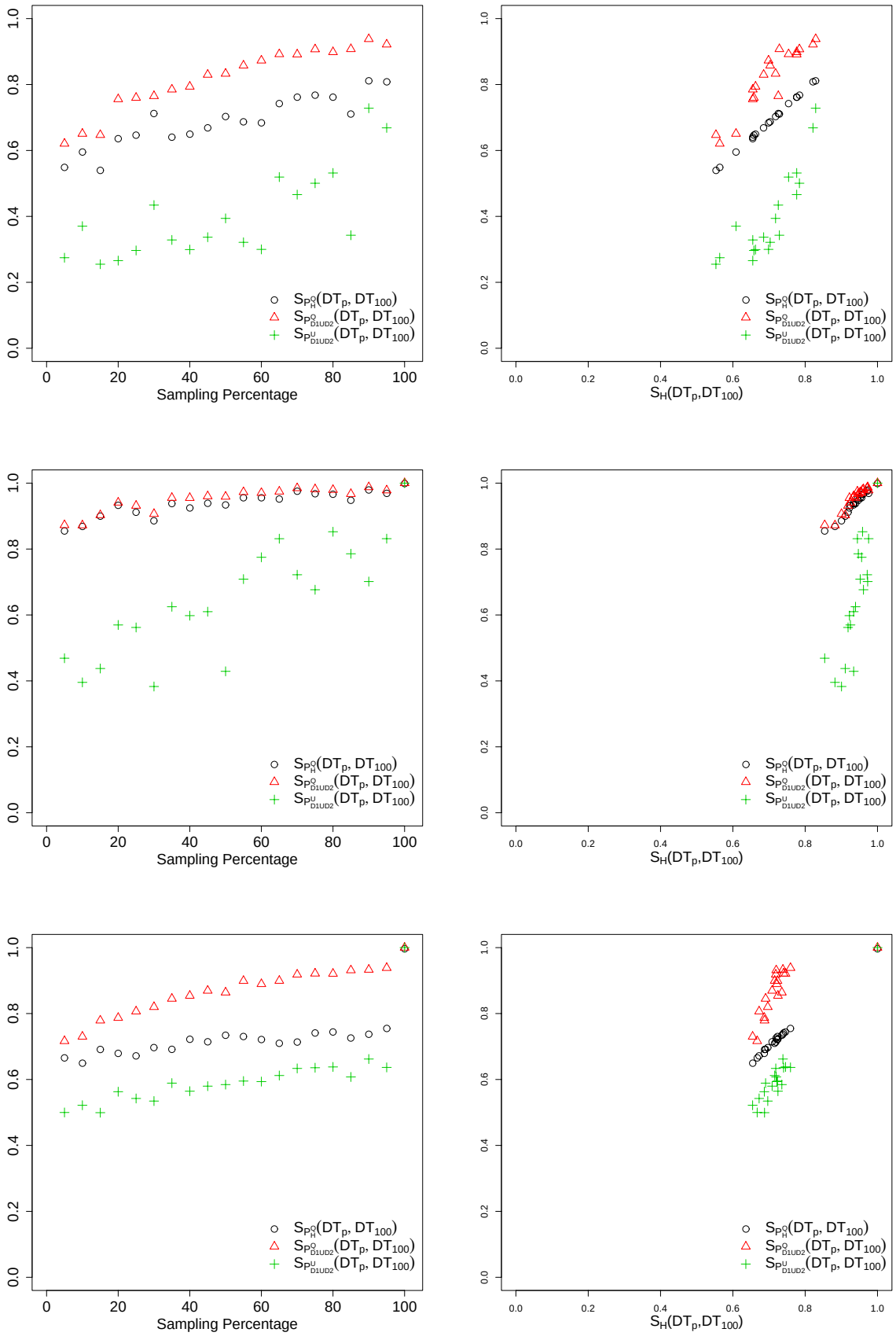

Figure 3: Evolution of the DTs semantic similarity measures with the sampling rate (first column) and with S_H (second column) for datasets: mfeat-fourier (top), segment-challenge (middle), waveform-5000 (bottom)

dataset	$S_{P^U_{D_1 \cup D_2}}$	$S_{P^Q_{D_1 \cup D_2}}$	$S_{P^Q_H}$
mfeat-factors	0.692	0.971	0.993
mfeat-fourier	0.852	0.927	0.999
mfeat-karhunen	0.858	0.910	0.999
mfeat-zernike	0.869	0.911	0.987
segment-challenge	0.831	0.951	0.986
waveform-5000	0.969	0.712	0.998

Table 2: Correlation coefficient of $S_{P^Y_X}(DT_p, DT_{100})$ with $S_H(DT_p, DT_{100})$

dataset	$S_{P^U_{D_1 \cup D_2}}$	$S_{P^Q_{D_1 \cup D_2}}$	$S_{P^Q_H}$
mfeat-factors	0.504	0.063	0.014
mfeat-fourier	0.301	0.114	0.015
mfeat-karhunen	0.279	0.158	0.013
mfeat-zernike	0.316	0.108	0.022
segment-challenge	0.289	0.016	0.005
waveform-5000	0.120	0.140	0.003
Average	0.302	0.1	0.012

Table 3: Mean absolute deviation of $S_{P^Y_X}(DT_p, DT_{100})$ with $S_H(DT_p, DT_{100})$

probability distribution $P(A)$ governing the attribute space. The overlayed tree provides a partition of the full attribute space, the agreement or disagreement of the two decision trees in a given segment of that partition is more or less important depending on the density of that region under $P(A)$. If, for example, the two DTs disagree on a given region this is not going to affect their similarity, even if the volume of the region is large, as far as the probability density of that region under $P(A)$ is zero. Alternatively, if we do not want to assume a specific attribute space distribution and we want to compute similarity under all possible worlds we should make the assumption of a uniform distribution on the attribute space, a case that is also covered by our framework.

In order to evaluate our semantic similarity measures, we used the semantic similarity empirically estimated on a separate hold-out set. The performance of the different instantiations of the semantic similarity measures depends on how different was the estimation of $P(A)$ used in them from the $P(A)$ governing the hold-out set, on which the semantic similarity was computed. In fact, the choice of the appropriate $P(A)$ should be done based on the knowledge of the application domain. If we know that our learning problem is governed by a specific $P(A)$, then it is that $P(A)$ that should be "plugged" in the DT similarity measure. Alternatively, if no such knowledge exists, we can estimate $P(A)$ from the datasets from which the DTs were constructed, as it was done in the $S_{P^Q_{D_1 \cup D_2}}$ semantic similarity measure.

5 Conclusions and Future Work

In this paper we presented a general framework for the estimation of similarities between DTs and datasets, within a classification problem setting. We employ the DT models in order to compute either their semantic similarity or the similarity of the datasets that were used for their induction. The DT similarity is computed in terms of the agreement of the class predictions they return over the attribute space and, it corresponds to the DT semantic similarity. The computation of dataset similarity can be done on the basis of their attribute space probability distribution $P(A)$, their attribute-class joint probability distribution $P(A, C)$ or their attribute conditional class probability distribution $P(C|A)$. All the above comprise special cases of our framework.

Previous efforts have focused on comparing datasets, either with respect to $P(A, C)$ [3], or with respect to $P(C|A)$ [5]. On the other hand, in this paper, we focused on the estimation of the semantic similarity between DTs, i.e., the degree to which the DTs agree in their predictions over the attribute space. To the best of our knowledge, this is the first work towards this aim. The critical point in the computation of the DT similarity is the selection of an appropriate attribute space probability distribution $P(A)$ under which the computation will take place. This choice reflects our belief about the real world on which the DTs would be applied. If no prior knowledge exists, we could simply select the uniform distribution $U(A)$ for $P(A)$ and thus we would examine the DT similarity over all possible input worlds.

We experimented with different ways of estimating the attribute space probability distribution $P(A)$ and we compared the resulting instantiations of the DT semantic similarity measure with the actual semantic similarity, as this was established by the application of the DTs on an independent hold-out set. Depending on the knowledge we have about the $P(A)$ distribution that governs the independent hold-out set, the computed DT semantic similarity is a more or less good predictor of the actual semantic similarity. More specifically, when $P(A)$ is computed by querying the actual datasets, the corresponding DT similarity $S_{P^Q_{D_1 \cup D_2}}$ is a very good predictor of the true semantic similarity. Actually, we expect the value of $S_{P^Q_{D_1 \cup D_2}}$ to converge to the real value of the semantic similarity as the size of the datasets increases, since the estimated $P(A)$ will converge to the true $P(A)$.

We believe that the greatest contribution of a DT semantic similarity measure is the potential that it offers to determine whether the observed differences are simply superficial structural differences or they reflect real semantic differences on the described concepts, and moreover, to quantify these differences - this is a problem that "deplores" DTs due to their high sensitivity to training dataset changes.

The provision of a semantic similarity measure for classification models, here decision trees, allows us to perform a number of standard mining tasks that are based on similarity/distance measures, not on the raw data anymore, but rather on the classification models extracted from these raw data, i.e., *meta–mining*. For example, using the semantic similarity measure we can cluster DTs and compute a *representative DT* for each cluster. A typical application of that could be the simplification of *ensembles* of decision trees, such as the ones produced by boosting, bagging and random forests, where only the prototype decision tree of each cluster is retained. Another alternative to the simplification of DT ensembles, that does not make use of the semantic similarity measure, is the construction of the overlayed tree from all the component DTs of the ensemble. Each region of the overlayed tree will be labeled according to the labels of the corresponding regions of the original trees. The overlayed tree will have the same predictive power as the ensemble, since it will make exactly the same predictions, however its partitions will be much finer than the partitions of the original trees thus having a larger complexity than its constituents. Nevertheless, it is possible to simplify the overlayed DT by applying standard pruning techniques. The apparent advantage of having a single DT, or a small set of DTs, instead of the full ensemble is the much easier interpretation of the learned model.

Also, in the ensemble research, a lot of work has been done on measuring the diversity of the base classifiers, since accuracy improvement can be achieved only if the base classifiers are sufficiently diverse ([1], [6]). Several measures have been proposed in order to estimate deviation, e.g., error correlation. The proposed semantic similarity measure could be also employed towards this aim.

The idea of a representative DT for a set of DTs could be also useful in a *classification error estimation* scenario. Typically, in error estimation a re-sampling technique is applied resulting in a number of different models, the final result is an estimation of the classification performance of the algorithm and not that of a single tree. The question is which model to choose among the different models that were produced; one solution would be to choose the median model, i.e., the one that abstains the smaller distance from all the other models.

Finally, there are several extensions/ improvements over the basic framework. In the current version, we restrict on continuous predictive attributes, however categorical attributes should be also considered. Also, in case of the pattern dependent probability estimation (Equations 3.8, 3.9), we adopt the assumption that instances are uniformly distributed within each region. We plan to release this assumption by employing some density approximation technique like histograms.

Acknowledgment Irene Ntoutsi is partially supported by the "Heracletos"program co-funded by the European Social Fund and national resources (Operational Program for Educational and Vocational Training II - EPEAEK II).

References

[1] K. Ali, and M Pazzani. Error Reduction through Learning Multiple Descriptions. *Machine Learning*, 24(3):173–202, 1996.

[2] L. Breiman, J. Friedman, R. Olshen and C. Stone. *Classification and Regression Trees*. Wadsworth International, 1984.

[3] V. Ganti, J. Gehrke and R. Ramakrishnan. A framework for measuring changes in data characteristics. In *PODS*, pages 126–137, 1999.

[4] A. Jain, R. Duin and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[5] I. Pekerskaya, J. Pei and K. Wang. Mining changing regions from access-constrained snapshots: A cluster-embedded decision tree approach. *Journal of Intelligent Information Systems (special Issue on Mining Spatio-Temporal Data)*, 2007.

[6] K. Tumer and J. Ghosh. Classifier combining: analytical results and implications. In *AAAI -Workshop in Induction of Multiple Learning Models*, pages 126–137, 1995.

[7] P. Turney. Technical note: Bias and the quantification of stability. *Machine Learning*, 20:23–33, 1995.

[8] UCI. Machine learning repository. *http://www.ics.uci.edu/ mlearn/MLSummary.html*.

[9] H. Wang and J. Pei. A random method for quantifying changing distributions in data streams. In *PKDD*, pages 684–691, 2005.

[10] K. Zhang and D. Shasha. Fast algorithms for the editing distance between trees and related problems. *SIAM, Journal of Computation*, 18:1245–1262, 1989.