Mining Diversified Shared Decision Tree Sets for Discovering Cross Domain Similarities

Guozhu Dong and Qian Han

Knoesis Center, and Department of Computer Science and Engineering,
Wright State University, Dayton, Ohio, USA
{guozhu.dong,han.6}@wright.edu

Abstract. This paper studies the problem of mining diversified sets of shared decision trees (SDTs). Given two datasets representing two application domains, an SDT is a decision tree that can perform classification on both datasets and it captures class-based population-structure similarity between the two datasets. Previous studies considered mining just one SDT. The present paper considers mining a small diversified set of SDTs having two properties: (1) each SDT in the set has high quality with regard to "shared" accuracy and population-structure similarity and (2) different SDTs in the set are very different from each other. A diversified set of SDTs can serve as a concise representative of the huge space of possible cross-domain similarities, thus offering an effective way for users to examine/select informative SDTs from that huge space. The diversity of an SDT set is measured in terms of the difference of the attribute usage among the SDTs. The paper provides effective algorithms to mine diversified sets of SDTs. Experimental results show that the algorithms are effective and can find diversified sets of high quality SDTs.

Keywords: Knowledge transfer oriented data mining, research by analogy, shared decision trees, cross dataset similarity, shared accuracy similarity, matching data distribution similarity, tree set diversity.

1 Introduction

Shared knowledge structures across multiple domains play an essential role in assisting users to transfer understanding between applications and to perform analogy based reasoning and creative thinking [3,6,9,8,10], in supporting users to perform research by analogy [4], and in assessing similarities between datasets in order to avoid negative learning transfer [16]. Motivated by the above, Dong and Han [5] studied the problem of mining knowledge structures shared by two datasets, with a focus on mining a single shared decision tree. However, providing only one shared decision tree may present only a limited view of shared knowledge structures that exist across multiple domains and does not offer users a concise representative of the space of possible shared knowledge structures. Moreover, computing all possible shared decision trees is infeasible. The purpose of this paper is to overcome the above limitations by studying the problem of mining diversified sets of shared decision trees across two application domains.

V.S. Tseng et al. (Eds.): PAKDD 2014, Part II, LNAI 8444, pp. 534–547, 2014.

In a diversified set of shared decision trees, each individual tree is a high quality shared decision tree in the sense that (a) the tree has high accuracy in classifying data for each dataset and the tree has high cross-domain class-distribution similarity at all tree nodes, and (b) different trees are structurally highly different from each other in the sense that they use very different sets of attributes. The requirements in (a) will ensure that each shared decision tree captures high quality shared knowledge structure between the two given datasets, providing the benefit that each root-to-leaf path in the tree corresponds to a similar rule (having similar support and confidence) for the two datasets and the benefit that the tree nodes describe similar data populations in the two datasets connected by similar multi-node population relationships.

Presenting too many shared decision trees to human users will imply that users will need to spend a lot of time to understand those trees in order to select the ones most appropriate for their application. Efficient algorithms solving the problem of mining diversified sets of shared decision trees meeting the requirements in (b) can offer a *small representative* set of high quality shared decision trees that can be understood without spending a lot of time, hence allowing users to more effectively select the tree most appropriate for their situation. Figure 1 illustrates the points given above, with the six stars as the diversified representatives of all shared decision trees.

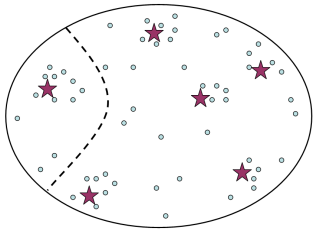


Fig. 1. Diversified Representatives of SDT Space

The main contributions of this paper include the following: (1) The paper motivates and formulates the diversified shared decision tree set problem. (2) It presents two effective algorithms to construct diversified high quality shared decision tree sets. (3) It reports an extensive experimental evaluation on the diversified shared decision tree set mining algorithms. (4) The shared decision trees reported in the experiments are mined from high dimensional microarray datasets for cancers, which can be useful to medical researchers.

The rest of the paper is organized as follows. Section 2 summarizes related work. Section 3 formally introduces the diversified shared decision tree set problem and associated concepts. Sections 4 presents our algorithms for mining diversified shared decision tree sets. Section 5 reports our experimental evaluation. Section 6 summarizes the results and discusses several future research problems.

2 Related Work

Limited by space, we focus on previous studies in four highly related areas.

Importance of Similarity/Analogy from Psychology and Cognitive Science: Psychology/cognitive science studies indicate that analogy plays a vital role in human thinking and reasoning, including *creative thinking*. For example, Fauconnier [6] states that "*Our conceptual networks are intricately structured by*

analogical and metaphorical mappings, which play a key role in the synchronic construction of meaning ...". Gentner and Colhoun [9] state that "*Much of humankind's remarkable mental aptitude can be attributed to analogical ability.*" Gentner and Markman [10] suggest "that both similarity and analogy involve a process of structural alignment and mapping." Christie and Gentner [3] suggest, based on psychological experiments, that "structural alignment processes are crucial in developing new relational abstractions" and *forming new hypothesis.*

Learning Transfer: In learning transfer [16], it is typical to use available structure/knowledge of an auxiliary application domain to help build better classifiers/clusters for a target domain where there is a lack of data with class label or other domain knowledge. The constructed classifiers are not intended to capture cross-domain similarities. In contrast, our work focuses on mining (shared) knowledge structures to capture cross domain similarity; this is a key difference between learning transfer and our work. One of the intended uses of our mining results is for direct human consumption. Our mining results can also be used to assess cross domain similarity, to help avoid negative transfer where learning transfer actually leads to poorer results, since learning transfer is a process that is based on utilizing cross domain similarities that exist.

Shared Knowledge Structure Mining: Reference [4] defined the *cross domain similarity mining* (CDSM) problem, and motivated CDSM with several potential applications. CDSM has big potential in (1) supporting understanding transfer and (2) supporting research by analogy, since similarity is vital to understanding/meaning and to identifying analogy, and since analogy is a fundamental approach frequently used in hypothesis generation and in research. CDSM also has big potential in (3) advancing learning transfer since cross domain similarities can shed light on how to best adapt classifiers/clusterings across given domains and how to avoid negative transfer. CDSM can also be useful for (4) solving the schema/ontology matching problem. Reference [5] motivated and studied the shared decision tree mining problem, but that paper focused on mining just one shared decision tree. Several concepts of this paper were borrowed from [5], including shared accuracy, data distribution similarity, weight vector pool (for two factors), and information gain for two datasets. We enhance that paper by considering mining diversified sets of shared decision trees.

Ensemble Diversity: Much has been done on using ensemble diversity among member classifiers [14] to improve ensemble accuracy, including data based diversity approaches such as Bagging [2] and Boosting [7]. However, most previous studies in this area focused on classification behavior diversity, in the sense that different classifiers make highly different classification predictions. Several studies used attribute usage diversity to optimize ensemble accuracy, in a less systematic manner, including the random subspace method [13] and the distinct tree root method [15]. Our work focuses on attribute usage diversity aimed at providing diversified set of shared knowledge structures between datasets. The concept of attribute usage diversity was previously used in [12], to improve classifier ensemble diversity and classification accuracy for just one dataset.

3 Problem Definition

In this section, we introduce the problem of mining diversified sets of shared decision trees. To that end, we also need to define a quality measure on diversified sets of shared decision trees, which is based on the quality of individual shared decision trees and on the diversity measure for sets of shared decision trees.

As mentioned earlier, a shared decision tree (SDT) for a given dataset pair $(D_1 : D_2)$ is a decision tree that can classify both data in D_1 and data in D_2.

We assume that D_1 and D_2 are datasets with (1) an identical set of class names and (2) an identical set[1] of attributes. Our aim is to mine a small diversified set of high quality SDTs with these properties: (a) each tree in the set (1) is highly accurate in each D_i and (2) has highly similar data distribution in D_1 and D_2, and (b) different trees in the set are highly different from each other.

3.1 Shared Accuracy and Data Distribution Similarity of SDT Set

The shared accuracy and data distribution similarity measures for shared decision tree (SDT) sets are based on similar measures defined for individual SDTs [5], which we will review below. Let T be an SDT for a dataset pair $(D_1 : D_2)$, and let $Acc_{D_i}(T)$ denote T's accuracy[2] on D_i.

Definition 1. The *shared accuracy* of T (denoted by $\mathsf{SA}(T)$) is defined as the minimum of T's accuracies on D_1 and D_2: $\mathsf{SA}(T) = \min(Acc_{D_1}(T), Acc_{D_2}(T))$.

The *data distribution similarity* of T reflects population-structure (or class distribution) similarity between the two datasets across the nodes of T. The *class distribution vector* of D_i at a tree node V is defined by

$$CDV_i(V) = (Cnt(C_1, SD(D_i, V)),\ Cnt(C_2, SD(D_i, V))),$$

where $Cnt(C_j, SD(D_i, V)) = |\{t \in SD(D_i, V) \mid t\text{'s class is } C_j\}|$, and $SD(D_i, V)$ is the subset of D_i for V (satisfying the conditions on the root-to-V path). The *distribution similarity* (DSN) at V is defined as $DSN(V) = \frac{CDV_1(V) \cdot CDV_2(V)}{\|CDV_1(V)\| \cdot \|CDV_2(V)\|}$.

Definition 2. The *data distribution similarity* of an SDT T over $(D_1 : D_2)$ is defined as $\mathsf{DS}(T) = avg_V DSN(V)$, where V ranges over nodes of T.

We can now define SA and DS for shared decision tree sets.

Definition 3. Let TS be a set of SDTs over dataset pair $(D_1 : D_2)$. The *shared classification accuracy* of TS is defined as $\mathsf{SA}(TS) = avg_{T \in TS}\mathsf{SA}(T)$, and the *data distribution similarity* of TS is defined as $\mathsf{DS}(TS) = avg_{T \in TS}\mathsf{DS}(T)$.

[1] If D_1 and D_2 do not have identical classes and attributes, one will need to identify an 1-to-1 mapping between the classes of the two datasets, and an 1-to-1 mapping between the attributes of the two datasets. The 1-to-1 mappings can be real or hypothetical (for "what-if" analysis) equivalence relations on the classes/attributes.

[2] When D_i is small, one may estimate $Acc_{D_i}(T)$ directly using D_i. Holdout testing can be used when the datasets are large.

3.2 Diversity of SDT Set

To define the *diversity of SDT sets*, we need to define tree-pair difference, and we need a way to combine the tree-pair differences for all possible SDT pairs.[3]

We measure the difference between two SDTs in terms of their attribute usage summary (AUS). Let $A_1, A_2, \ldots, A_n$ be a fixed list enumerating all shared attributes of D_1 and D_2.

Definition 4. The *level-normalized-count* AUS ($\mathsf{AUS}_{\mathsf{LNC}}$) of an SDT T over $(D_1 : D_2)$ is defined as

$$\mathsf{AUS}_{\mathsf{LNC}}(T) = (\frac{Cnt_T(A_1)}{avgLvl_T(A_1)}, \ \ldots, \ \frac{Cnt_T(A_n)}{avgLvl_T(A_n)}),$$

where $Cnt_T(A_i)$ denotes the number of occurrences of attribute A_i in T, and $avgLvl_T(A_i)$ denotes the average level of $A_i's$ occurrences in T.

The root is at level 1, the children of the root are at level 2, and so on. In the $\mathsf{AUS}_{\mathsf{LNC}}$ measure, nodes near the root have high impact since those nodes have small level number and attributes used at those nodes often have small $avgLvl_T$.

One can also use the *level-listed count* ($\mathsf{AUS}_{\mathsf{LLC}}$) approach for AUS. Here we use a matrix in which each row represents the attribute usage in one tree level: Given an SDT T with L levels, for all attributes A_i and integers l satisfying $1 \leq l \leq L$, the (l, i) component of $\mathsf{AUS}_{\mathsf{LLC}}$ has the value $Cnt_T(l, A_i)$ (the occurrence frequency count for attribute A_i in the l^{th} level of T).
Remark: $\mathsf{AUS}_{\mathsf{LNC}}$ pays more attention to nodes near the root, while $\mathsf{AUS}_{\mathsf{LLC}}$ gives more emphasis to levels near the leaves (there are many nodes at those levels).

Definition 5. Given an attribute usage summary measure AUS_μ, the *tree pair difference* (TPD) for two SDTs T_1 and T_2 is defined as

$$\mathsf{TPD}_\mu(T_1, T_2) = 1 - \frac{\mathsf{AUS}_\mu(T_1) \cdot \mathsf{AUS}_\mu(T_2)}{\|\mathsf{AUS}_\mu(T_1)\| \cdot \|\mathsf{AUS}_\mu(T_2)\|}.$$

We can now define the SDT set diversity concept.

Definition 6. Given an SDT set TS and an AUS measure AUS_μ, the *diversity* of TS is defined as $\mathsf{TD}_\mu(TS) = avg\{\mathsf{TPD}_\mu(T_i, T_j) \mid T_i, T_j \in TS, and\ i \neq j\}$.

3.3 Diversified Shared Decision Tree Set Mining Problem

To mine desirable diversified SDT sets, we need an objective function. This section defines our objective function, which combines the quality of the SDTs and the diversity among the SDTs.

Definition 7. Given an attribute usage summary method AUS_μ, the quality score of an SDT set TS is defined as:

$$\mathsf{SDTSQ}_\mu(TS) = min(\mathsf{SA}(TS), \mathsf{DS}(TS), \mathsf{TD}_\mu(TS)) * avg(\mathsf{SA}(TS), \mathsf{DS}(TS), \mathsf{TD}_\mu(TS)).$$

[3] We borrow the diversity concepts from [12], which considered mining diversified decision tree ensembles for one dataset.

We also considered other definitions using e.g. average, weighted average, and the harmonic mean of the three factors. They were not selected, since they give smaller separation of quality scores or require parameters from users. The above formula is chosen since it allows each of SA, DS, and TD to play a role, it is simple to compute, and it does not require any parameters from users.

We now turn to defining the diversified SDT set mining problem.

Definition 8 (Diversified Shared Decision Tree Set Mining Problem). Given a dataset pair *(D_1:D_2)* and a positive integer k, the *diversified shared decision tree set mining problem (*KSDT*)* is to mine a diversified set of k SDTs with high SDTSQ from the dataset pair.

An example SDT set mined from two cancer datasets will be given in § 5.

4 KSDT-Miner

This section presents two KSDT mining algorithms, for mining diversified high quality SDT sets. One is the parallel KSDT-Miner (PKSDT-Miner), which builds a tree set concurrently and splits one-node-per-tree in a round-robin fashion; the other one is the sequential KSDT-Miner (SKSDT-Miner), which mines a tree set by building one complete tree after another.

In comparison, PKSDT-Miner gives *all SDTs* almost equal opportunity[4] in selecting desirable attributes for use in high impact nodes near the roots of SDTs, whereas SKSDT-Miner gives SDTs built earlier more possibilities in selecting desirable attributes (even for use at low impact nodes near the leaves), which deprives the chance of later SDTs in using those attributes at high impact nodes.

Limited by space and due to the similarity in most ideas except the node split order, we present PKSDT-Miner and omit the details of SKSDT-Miner.

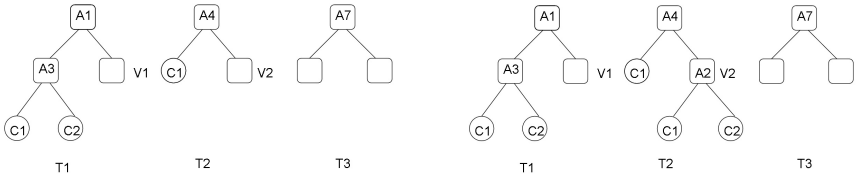
4.1 Overview of PKSDT-Miner

PKSDT-Miner builds a set of SDTs in parallel, in a node-based round-robin manner. In each of the round-robin loop, the trees are processed in an ordered manner; for each tree, one node is selected and spilt. Figure 2 illustrates with two consecutive states in such a loop: 2(a) gives three (partial) trees (blank rectangles are nodes to be split), and 2(b) gives those trees after splitting node V_2 of T_2. Here, PKSDT-Miner splits node V_2 in T_2 even though V_1 in T_1 can be split. PKSDT-Miner will select a node in T_3 to split next.

4.2 Aggregate Tree Difference

To build highly diversified tree sets, the *aggregate tree difference* (ATD) is used to measure the differences between a new/modified tree T and the set of other trees TS. One promising approach is to define ATD as the average of ℓ smallest

[4] An attribute may be highly desirable in more than one tree.



(a) State p (b) State p+1

Fig. 2. PKSDT-Miner Builds Trees in Round-robin Manner

$\mathsf{TPD}_\mu(T, T')$ values ($T' \in TS$). We define a new aggregation function called $avgmin_\ell$, where $avgmin_\ell(S)$ is the average of the ℓ smallest values in a set S of numbers. (In experiments our best choice for ℓ was 3.) Then the *μ-ℓ-minimal aggregate tree difference* ($\mathsf{ATD}_{\mu,avgmin\ell}$) is defined as the average TPD_μ between T and the ℓ most similar trees in TS:

$$\mathsf{ATD}_{\mu,avgmin\ell}(T, TS) = avgmin_\ell(\{\mathsf{TPD}_\mu(T, T') | T' \in TS\}).$$

PKSDT-Miner selects an attribute and a value to split a tree node by maximizing an objective function IDT that combines *information gain* (to be denoted by IG and defined below) and *data distribution similarity* (DS) on two datasets, and *aggregate tree difference* (ATD) between the current tree and the other trees. To tradeoff the three factors, they are combined using a weighted sum based on a weight vector $w = (w_{IG}, w_{DS}, w_{ATD})$ whose three weights are required to satisfy $0 < w_{IG}, w_{DS}, w_{ATD} < 1$ and $w_{IG} + w_{DS} + w_{ATD} = 1$.

Given an AUS measure μ, an aggregation method $\alpha \in \{avg, min, avgmin_\ell\}$, a tree T and a tree set TS, the *μ-α aggregate tree difference* is defined as

$$\mathsf{ATD}_{\mu,\alpha}(T, TS) = \alpha(\{\mathsf{TPD}_\mu(T, T') \mid T' \in TS\}).$$

For example, $\mathsf{ATD}_{\mu,min}(T, TS) = \min_{T' \in TS}(\mathsf{TPD}_\mu(T, T'))$ when α is min. Each variant of ATD can be used in our two SDT set mining algorithms, resulting a number of variant algorithms. For instance, the standard version of the PKSDT-Miner algorithm can be written as PKSDT-Miner(LNC,$avgmin\ell$) and we can replace LNC by LLC to get PKSDT-Miner(LLC,$avgmin\ell$).

4.3 IG for Two Datasets

This paper uses the union-based definition of IG for two datasets of [5]. ([5] discussed other choices and the union-based way was shown to be the best by experiments.) For each attribute A and split value a, and dataset pair $(D'_1 : D'_2)$ (associated with a given tree node), the *union-based information gain* is defined as $\mathsf{IG}(A, a, D'_1, D'_2) = \mathsf{IG}(A,\ a,\ D'_1 \cup\ D'_2)$. (The IG function is overloaded: IG in the LHS is 4-ary while IG in the RHS is 3-ary.) $\mathsf{IG}(A, a, D')$ is defined in terms of entropy, as used for decision tree node splitting.

4.4 The Algorithm

PKSDT-Miner has six inputs: Two datasets D_1 and D_2, a set $AttrSet$ of candidate attributes that can be used in shared trees, a dataset size threshold $MinSize$ for node-splitting termination, a weight vector w (w_{IG} on information gain, w_{DS} on data distribution similarity, w_{ATD} on aggregate tree difference), and an integer k for the desired number of trees. PKSDT-Miner calls PKSDT-SplitNode (Function 1) to split nodes for each tree.

Algorithm 1. PKSDT-Miner

Input: $(D_1 : D_2)$: Two datasets
$AttrSet$: Set of candidate attributes that can be used
$MinSize$: Dataset size threshold for splitting termination
$w = (w_{IG}, w_{DS}, w_{ATD})$: A weight vector on IG, DS, and ATD
k: Desired number of trees
Output: A diversified shared decision tree set TS for $(D_1 : D_2)$.
Method:
1. Create root node V_p for each tree T_p $(1 \leq p \leq k)$;
2. Repeat
3. For $p = 1$ to k do
4. let node V be the next node[a] of tree T_p to split;
5. Call PKSDT-SplitNode$(T, V, D_1, D_2, AttrSet, TS, MinSize, w)$;
6. Until there are no more trees with nodes that can be split[b];
7. Output the diversified shared decision tree set TS.

[a] The next node of a tree to split is determined by a tree traversal method, which can be depth first, breadth first, and so on. We use depth first here.

[b] No more node to split means that all candidate split nodes satisfy the termination conditions defined in Function ShouldTerminate.

PKSDT-SplitNode splits the data of a node V of a tree T by picking the split attribute and split value that optimize the IDT score. Let T be the tree that we wish to split, and let TS be the other trees that we have built. Let V be T's current node to split, and A and a_V be resp. a candidate splitting attribute/value. Let $T(A, a_V)$ be the tree obtained by splitting V using A and a_V. Then the IDT scoring function is defined by:

$$IDT(T(A, a_V), TS) = w_{IG} * IG(A, a_V) + \\ w_{DSN} * DSN(A, a_V) + w_{ATD} * ATD(T(A, a_V), TS),$$

where $IG(A, a_V)$ is the information gain for V when split by A and a_V, $DSN(A, a_V)$ is the average DSN value of the two children nodes of V.

Function ShouldTerminate determines if nodes splitting should terminate. (Our algorithms aim to build simple trees and avoid "overfitting".) It uses two techniques. (1) When many attribtues are available, we restrict the candidate attributes to those whose IG is ranked high in both datasets, so avoiding non-discriminative attributes that are locally discriminative at a given node. (2) We stop splitting for a given tree node when at least one dataset is small or pure.

Function 1. PKSDT-SplitNode($T, V, D'_1, D'_2, AttrSet, TS, MinSize, w$)

1. If ShouldTerminate($V, D'_1, D'_2, MinSize, AttrSet$) then assign the majority class in D'_1 and D'_2 as class label of V and return;
2. Select the attribute B and value b_V that maximize IDT, that is $IDT(T(B, b_V), TS)$=max$\{IDT(T(A, a_V), TS) \mid A \in AttrSet$, and a_V is a common candidate split value for A at $V\}$;
3. Create left child node V_l of V, with "$B \leq b_V$" as the corresponding edge's label, and let $D'_{il} = \{t \in D'_i \mid t$ satisfies "$B \leq b_V$"$\}$ for $i = 1, 2$;
4. Create right child node V_r of V, with "$B > b_V$" as the corresponding edge's label, and let $D'_{ir} = \{t \in D'_i \mid t$ satisfies "$B > b_V$"$\}$ for $i = 1, 2$.

4.5 Weight Vector Pools

Different dataset pairs have different characteristics concerning IG, DS and ATD. To mine the best SDT set, we need to treat different characteristics using appropriate focus/bias. (It is open if one can determine the characteristics of a dataset pair without performing SDT set mining.) We solve the problem by using a pool of weight vectors to help mine (near) optimal SDTs efficiently. Such a pool is a small representative set of all possible weight vectors.

We consider two possible weight vector pools: WVP_1 contains 36 weight vectors, defined by $WVP_1 = \{x \mid x$ is a multiple of 0.1 and $0 < x < 1\}$. WVP_2=$\{$(0.1,0.1,0.8),(0.1,0.3,0.6),(0.1,0.5,0.4),(0.1,0.7,0.2),(0.3,0.1,0.6), (0.3, 0.4, 0.3),(0.3,0.5,0.2),(0.5,0.1,0.4),(0.5,0.3,0.2),(0.7,0.2,0.1)$\}$. So WVP_2 contains 10 representative vectors selected from WVP_1. For each of the three factors, each pool contains some vectors where the given factor plays the dominant role.

5 Experimental Evaluation

This section uses experiments to evaluate KSDT-Miner, using real-world and also (pseudo) synthetic datasets. It reports that (1) PKSDT-Miner tends to build more diversified high quality SDT sets on average, which confirms the advantages of PKSDT-Miner analyzed in Section 4, and (2) KSDT-Miner is scalable w.r.t. number of tuples/attributes/trees. It discusses (3) how KSDT-Miner performs concerning the use of weight vectors. It also examines (4) how KSDT-Miner performs when it uses different AUS_μ and $\mathsf{ATD}_{\mu,\alpha}$ measures. Finally, it reports that KSDT-Miner outperforms SDT-Miner regarding mining one single SDT.

In the experiments, we set $\ell = 3$, $MinSize = 0.02 * min(|D_1|, |D_2|)$ and $AttrSet = \{A \mid rank_1(A) + rank_2(A)$ is among the smallest 20% of all shared attributes, where $rank_i(A)$ is the position of A when D_i's attributes are listed in decreasing IG order$\}$. Experiments were conducted on a 2.20 GHz AMD Athlon with 2 GB memory running Windows XP, with codes implemented in Matlab. To save space, the tables may list results on subsets of the 15 dataset pairs on the 6 microarray datasets, although the listed averages are for all 15 pairs.

5.1 Datasets and Their Preprocessing

Our experiments used six real-world microarray gene expression datasets for cancers.[5] ArrayTrack [20] was used to identify shared (equivalent) attributes. Two genes are shared if they represent the same gene in different gene name systems. Table 1 lists the number of shared attributes for the 15 dataset pairs.

Table 1. Number of Shared Attributes (NSA) between Dataset Pairs

Dataset Pair	NSA
(BC:CN), (BC:DH), (BC:LM)	5114
(CN:DH), (CN:LM), (DH:LM)	7129
(CN:LB), (DH:LB), (LB:LM)	5313
(CN:PC), (DH:PC), (LM:PC)	5317
(BC:LB)	8123
(BC:PC)	8124
(LB:PC)	9030

In some dataset pairs the two datasets have very different class ratios. The class ratio of a dataset is likely an artifact of the data collection process and may not have practical implications. However, class ratio difference can make it hard to compare quality values for results mined from different dataset pairs. To address this, we use sampling with replacement method to replicate tuples so that class ratios for the two datasets are nearly the same.

5.2 Example Diversified SDT Set Mined from (DH:LM)

We now give[6] an example diversified set of two shared decision trees mined from real (cancer) dataset pair (DH:LM) in Figures 3 and 4. For each tree, data in two datasets have very similar distributions at tree nodes (the average DSN for each tree is about 0.977) and the leaf nodes are very pure with average shared classification accuracy of leaf nodes being about 0.963. For the diversified tree set $\{T_1, T_2\}$, tree diversity is 1 since these two trees don't share any splitting attributes, and the SDTSQ is about 0.944.

5.3 KSDT-Miners Mine Diversified High Quality SDT Sets

Experiments show that KSDT-Miners are able to mine diversified high quality SDT sets. Table 2 lists the statistics of best SDT sets mined by either PKSDT-Miner or SKSDT-Miner from each of the 15 dataset pairs in Table 1 (using all weight vectors). For the 15 dataset pairs, PKSDT-Miner got the best SDT sets in 9 pairs, and SKSDT-Miner got the best in 6 pairs. We include the result for (BC: LM) to indicate that it is not possible to always have high quality SDTs (as expected).

[5] References for the datasets: *BC* (breast cancer) [21], *CN* (Central Nervous System) [17], *DH* (DLBCL-Harvard*) [18], *LB* (Lung Cancer-BAWH) [11], *LM* (Lung Cancer-Michigan*) [1], *PC* (Prostate Cancer) [19].

[6] We draw shared decision tree figures as follows: For each node V , we show $CDV_1(V)$ for D_1 at $V's$ left, show $CDV_2(V)$ for D_2 at $V's$ right, and show $DSN(V)$ below V.

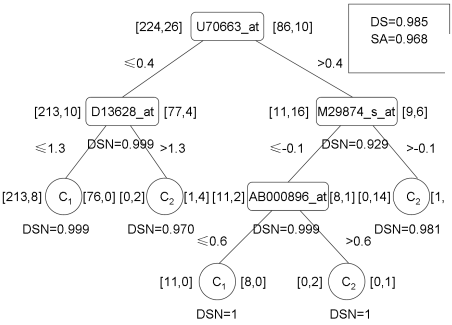


Fig. 3. Shared Decision Tree T_1

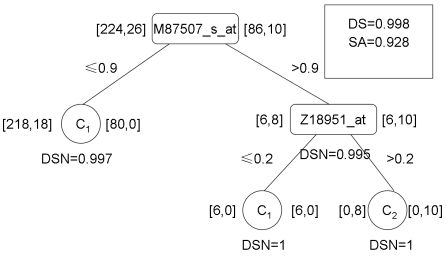


Fig. 4. Shared Decision Tree T_2

Table 2. Stats of Best SDT Sets

Dataset Pair	DS	SA	TD	SDTSQ
(BC: CN)	0.98	0.98	0.99	0.96
(BC: DH)	0.98	0.97	1	0.95
(BC: LB)	0.97	0.97	1	0.95
(BC: LM)	0.94	0.74	1	0.67
(BC: PC)	0.97	0.97	1	0.95
(CN: PC)	0.98	0.98	0.99	0.96
Average*	0.96	0.95	0.99	0.92

Table 3. PKSDT(P) vs SKSDT(S)

Dataset Pair	TD P	TD S	SDTSQ P	SDTSQ S
(BC: CN)	0.97	0.97	0.93	0.93
(BC: LB)	0.94	0.91	0.88*	0.85
(BC: LM)	0.92	0.92	0.61	0.63*
(BC: PC)	0.95	0.94	0.89	0.88
(CN: LB)	0.98	0.97	0.93	0.92
(DH: LB)	0.96	0.95	0.91*	0.88
Average*	0.94	0.93	0.86	0.85

5.4 Comparison between PKSDT-Miner and SKSDT-Miner

Experiments show that PKSDT-Miner is better than SKSDT-Miner. Indeed, PKSDT-Miner gets SDT sets of higher quality values on average, albeit slightly, and it never gets SDT sets of lower quality values (see Table 3, which gives the average TD and SDTSQ values for best diversified tree sets mined by PKSDT-Miner and SKSDT-Miner respectively when using all weight vectors). As noted for Table 2, PKSDT-Miner got the best SDT sets in 9, whereas SKSDT-Miner got the best in only 6, out of the 15 dataset pairs. Below we only consider PKSDT-Miner.

5.5 Comparison of AUS and ATD Variants

Experimental results demonstrate that (1) $\mathsf{AUS}_{\mathsf{LNC}}$ produces better results than $\mathsf{AUS}_{\mathsf{LLC}}$, and (2) $\mathsf{ATD}_{\mu,avgmin\ell}$ outperforms $\mathsf{ATD}_{\mu,min}$ (reason: when it is used a highly similar outlier may give too much influence) and $\mathsf{ATD}_{\mu,avg}$ (reason: when it is used the highly dissimilar cases may give big influence). The details are omitted to save space.

5.6 Weight Vector Issues

We examined the "best" and "worst" (w_{IG},w_{DS},w_{ATD}) weight vectors, which produce the SDT sets with the highest and lowest SDTSQ mined by PKSDT-Miner(LNC,$avgmin\ell$). (1) We observed that the average relative improvement of

the "best" over the "worst" is an impressive 4.8% and the largest is 20.5%. This indicates that the choice of weight vector has significant impact on the tree set quality mined by KSDT-Miner. (2) We also saw that no single weight vector is the best weight vector for all dataset pairs. This reflects the fact that different dataset pairs have different characteristics regarding which of IG, DS and ATD is most important.

Regarding which weight vectors may be better suited for which kinds of dataset pairs, we observed that there are three cases. (A) For some dataset pairs (e.g. (LM:PC)), weight vectors with high IG weight (and low DS weight, low ATD weight) tend to yield SDT sets with high SDTSQ. (B) For some dataset pairs (e.g. (BC:DH)), weight vectors with high DS weight tend to yield SDT sets with high SDTSQ. (C) For some dataset pairs (e.g. (BC:CN)), weight vectors with high ATD weight tend to yield SDT sets with high SDTSQ.

Experiment showed that using multiple weight vectors leads to much better performance than using a single weight vector. Moreover, SDTSQ scores of best SDT sets obtained using WVP_2 are almost identical to those obtained using WVP_1. Since WVP_2 is smaller (having 10 weight vectors) than WVP_1 (having 36 weight vectors), WVP_2 is preferred since it requires less computation time.

5.7 KSDT-Miner Outperforms SDT-Miner on SDTQ

Both KSDT-Miner and SDT-Miner can be used to mine a single high quality SDT, by having KSDT-Miner return the best tree in the SDT set it constructs. Experiments show that KSDT-Miner gives better performance than SDT-Miner. Indeed, the average relative SDTQ improvement by KSDT-Miner over SDT-Miner for all dataset pairs is 13.8%. For some dataset pairs, the relative improvement is about 45.3%. Through more detailed comparison, the average relative improvement on DS by KSDT-Miner over SDT-Miner for all dataset pairs is 3.2%, and on SA is 5.4%. Clearly, better single SDT can be mined when tree set diversity is considered.

5.8 Scalability of KSDT-Miner

We experimented to see how KSDT-Miner's execution time changes when the number of tuples/attributes/trees increases. Experiments show that execution time increases roughly linearly. (The figure is omitted to save space.) The experiments used synthetic datasets obtained by replicating tuples with added random noises up to a bound given by P% of the maximum attribute value magnitude (in order to get a desired number N of tuples), and by attribute elimination.

5.9 Using Fewer Attributes Leads to Poor SDT Sets

Incidently, we compared the SDTSQ of SDT sets mined from real dataset pairs using all available attributes against those obtained from projected data using fewer attributes (e.g. 100). On average, SDTSQ using all attributes is about 34.2% better than SDTSQ using only the first 100 attributes.

6 Concluding Remarks and Future Directions

In this paper we motivated the diversified shared decision tree set mining problem, presented two algorithms of KSDT-Miner, and evaluated the algorithms using real microarray gene expression data for cancers and using synthetic datasets. Experimental results show that KSDT-Miner can efficiently mine high quality shared decision trees. Future research directions include mining other types of shared knowledge structures (including those capturing alignable differences, defined as shared knowledge structures that capture cross-domain similarities and cross-domain differences within the context of the similarities given elsewhere in the shared knowledge structures) and utilizing such mined results to solve various research and development problems in challenging domains.

References

1. Beer, D.G., et al.: Gene-expression profiles predict survival of patients with lung adenocarcinoma. Nature Medicine 8, 816–824 (2002)
2. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
3. Christie, S., Gentner, D.: Where hypotheses come from: Learning new relations by structural alignment. Journal of Cognition and Development 11(3), 356–373 (2010)
4. Dong, G.: Cross domain similarity mining: Research issues and potential applications including supporting research by analogy. ACM SIGKDD Explorations (June 2012)
5. Dong, G., Han, Q.: Mining accurate shared decision trees from microarray gene expression data for different cancers. In: International Conference on Bioinformatics and Computational Biology, BIOCOMP 2013 (2013)
6. Fauconnier, G.: Mappings in Thought and Language. Cambridge University Press (1997)
7. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: ICML, pp. 148–156 (1996)
8. Gentner, D.: Structure mapping: A theoretical framework for analogy. Cognitive Science 7, 155–170 (1983)
9. Gentner, D., Colhoun, J.: Analogical processes in human thinking and learning. In: Glatzeder, B., Goel, V., von Müller, A. (eds.) Towards a Theory of Thinking. On Thinking, vol. 2. Springer, Heidelberg (2010)
10. Gentner, D., Markman, A.B.: Structure mapping in analogy and similarity. American Psychologist 52(1), 45–56 (1997)
11. Gordon, G.J., et al.: Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. Cancer Research 62, 4963–4967 (2002)
12. Han, Q., Dong, G.: Using attribute behavior diversity to build accurate decision tree committees for microarray data. J. Bioinformatics and Computational Biology 10(4) (2012)
13. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)
14. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning 51(2), 181–207 (2003)

15. Li, J., Liu, H.: Ensembles of cascading trees. In: ICDM, pp. 585–588 (2003)
16. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22(10), 1345–1359 (2010)
17. Pomeroy, S.L., et al.: Prediction of central nervous system embryonal tumour outcome based on gene expression. Nature 415, 436–442 (2002)
18. Shipp, M.A., et al.: Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. Nature Medicine 8, 68–74 (2002)
19. Singh, D., et al.: Gene expression correlates of clinical prostate cancer behavior. Cancer Cell 1(2), 203–209 (2002)
20. Tong, W., et al.: ArrayTrack-Supporting toxicogenomic research at the FDA's National Center for Toxicological Research (NCTR). EHP Toxicogenomics 111(15), 1819–1826 (2003)
21. Van't Veer, L.J., et al.: Gene expression profiling predicts clinical outcome of breast cancer. Nature 415, 530–536 (2002)