Machine Learning (2019) 108:785–807
https://doi.org/10.1007/s10994-019-05781-7

Annotation cost-sensitive active learning by tree sampling

Yu-Lin Tsou[1] · Hsuan-Tien Lin[1]

Received: 28 February 2018 / Accepted: 20 September 2018 / Published online: 1 April 2019

Abstract
Active learning is an important machine learning setup for reducing the labelling effort of humans. Although most existing works are based on a simple assumption that each labelling query has the same annotation cost, the assumption may not be realistic. That is, the annotation costs may actually vary between data instances. In addition, the costs may be unknown before making the query. Traditional active learning algorithms cannot deal with such a realistic scenario. In this work, we study annotation cost-sensitive active learning algorithms, which need to estimate the utility and cost of each query simultaneously. We propose a novel algorithm, the cost-sensitive tree sampling algorithm, that conducts the two estimation tasks together and solve it with a tree-structured model motivated from hierarchical sampling, a famous algorithm for traditional active learning. Extensive experimental results using datasets with simulated and true annotation costs validate that the proposed method is generally superior to other annotation cost-sensitive algorithms.

Keywords Annotation cost-sensitive · Active learning · Clustering · Decision tree

1 Introduction

In many machine learning scenarios, vast quantities of unlabelled instances can be easily acquired, yet high-quality labels are costly to obtain. For example, in fields such as medicine (Liu 2004) or biology (King et al. 2004), a massive number of experiments and analyses are needed to label a single instance, whereas collecting samples is a relatively easy task. Active learning is a machine learning setup that allows the machines to “ask questions” to the labelling oracle strategically (Settles 2010) to reduce the labelling cost. In particular, given a budget of the labelling cost, active learning algorithms aim to create a set of labelled

Editors: Masashi Sugiyama, Yung-Kyun Noh.

✉ Hsuan-Tien Lin
htlin@csie.ntu.edu.tw

Yu-Lin Tsou
R03922164@ntu.edu.tw

[1] Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan

data with a sequence of labelling queries (questions) so that the labelled set carries sufficient information to train an accurate learning model.

Active learning algorithms generally work by measuring the utility of each unlabelled instance for the learning model. Uncertainty sampling algorithms, which query the instances that are the most uncertain to the learning model, are arguably the most fundamental family of active learning algorithms (Lewis and Gale 1994; Tong and Koller 2001; Holub et al. 2008). That is, the uncertainty of each instance is taken as a measure of its utility within uncertainty sampling algorithms. Another important family is representative sampling algorithms, which take the uncertainty and representativeness of each instance as the utility measure (Kang et al. 2004; Huang et al. 2010; Xu et al. 2003; Dasgupta and Hsu 2008). The representativeness is often calculated on basis of clustering of the unlabelled instances. As a concrete example of representative sampling, the hierarchical sampling algorithm forms clusters by hierarchical clustering and then queries the instances within uncertain clusters (Dasgupta and Hsu 2008).

For the algorithms introduced above, and actually for most existing active learning algorithms, it is assumed that the labelling cost of each query is uniform. That is, the costs for the oracle to label every instance are exactly the same. Nevertheless, this assumption might not be true in real-world scenarios. For an article classification problem, the labelling cost could be the time spent by an oracle (usually a human annotator) while deciding the label, which depends on the length of the text and the complexity of the language and can differ from article to article. Nonuniform costs can deteriorate existing active learning algorithms. For instance, articles that are confusing to the annotator may have higher labelling costs, but uncertainty sampling may suggest querying them. Then, with a fixed budget of the labelling cost, uncertainty sampling can only query a few instances, leading to a possibly less-accurate model. It is thus important to design active learning algorithms that are annotation-cost-sensitive (or labelling-cost-sensitive) and will be the main focus of this work. For simplicity, we will use the term *cost-sensitive active learning* to describe our focus, while noting that it should not be confused with other works that study prediction-cost-sensitive active learning (Huang and Lin 2016).

There are some variations in the setup of cost-sensitive active learning. In Margineantu (2005), the labelling costs for all data instances are assumed to be known before querying, whereas in Settles et al. (2008), the cost of a data instance can only be acquired after querying its label. We focus on the latter setup, which closely matches the real-world scenario of human annotation. In other words, in each query of our setup, both the cost and label of the queried instance are revealed, while others' costs and labels remain unknown. Existing works (Haertel et al. 2008; Tomanek and Hahn 2010) thus need to estimate both the utility and cost of each instance at the same time in the setup and choose the instances with a high utility and low cost.

In this paper, we improve the joint estimation of the utility and cost for cost-sensitive active learning with a tree-structured model. The model is inspired by hierarchical sampling (Dasgupta and Hsu 2008), which also forms a tree with each internal node representing a cluster of instances. The key idea behind hierarchical sampling is that instances within the same cluster are likely to share the same label (Seeger 2000; Chapelle et al. 2003). We extend the idea by assuming a smooth cost function, so that the cost of an instance should be similar with its neighbors'. On the basis of the extended idea, we propose the cost-sensitive tree sampling (CSTS) algorithm for cost-sensitive active learning, in which both the utilities and costs are estimated in the tree-structured clusters constructed by a revised decision tree algorithm. In contrast to the hierarchical sampling algorithm, CSTS builds the clusters in a top-down manner to better use the label information. CSTS achieves cost-sensitivity by including cost estimation in its procedure and querying on the basis of a carefully designed criterion that mixes both the utility and cost. Extensive experiments using real-world datasets with simulated costs demonstrate that CSTS can usually provide superior results in comparison with existing cost-sensitive active learning algorithms. Furthermore, for a real-world benchmark dataset with true annotation costs, CSTS is stably superior to existing algorithms. The results justify the validity of the proposed CSTS algorithms.

The remainder of this paper is organized as follows. Section 2 summarizes the related works. In Sect. 3 we introduce the background of CSTS in detail and present the algorithms. Experiment results are discussed in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Related work

There are two categories of active learning: stream-based and pool-based. Unlabelled data instances can be sampled from the actual distribution with low costs under stream-based active learning; in the mean time, the active learning algorithm should be able to decide immediately whether to query the label of a newly sampled data instance or not (Cohn et al. 1994). On the other hand, pool-based active learning (Settles 2010) assumes that there exists a pool of available unlabelled data instances, and the active learning algorithm can query the label of any data instance inside the pool until the cost of total queries exceeds the budget. In general, pool-based learning is a more realistic setup regarding real-world problems, which is also the category we focus on.

Different querying strategies have been proposed to solve pool-based active learning problems. They mostly follow several major approaches, such as uncertainty sampling, representative sampling, query-by-committee, information theoretic, etc. Regarding the connection with cost-sensitive active learning, we focus on two popular approaches: uncertainty sampling and representative sampling.

- **Uncertainty sampling** The idea of uncertainty sampling (Lewis and Gale 1994) is to query the label for the data instance with the highest uncertainty in the classifier. For instance, Tong and Koller (2001) proposes querying of the data instance that is the closest to the decision boundary in a support vector machine (SVM); Holub et al. (2008) selects data instances for querying on the basis of the entropy of the label probabilities from a probabilistic classifier. These algorithms assume that the trained classifier is already sufficiently good; therefore only fine-tuning around the decision boundary is needed.
- **Representative sampling** In representative sampling, algorithms select data instances considering both representativeness and informativeness by seeking a way to model the data distribution. Among them, the clustering structure of the data instances is widely used. In Kang et al. (2004), the data instances that are closest to the centroid of each cluster are queried before other selection criteria are used; Huang et al. (2010) measures the representativeness of each data instance from both the cluster structure of unlabelled data instances and the class assignments of labelled data, and Xu et al. (2003) clusters those data instances close to the decision boundary in an SVM, and queries the labels of data instances near the center of each cluster; In Nguyen and Smeulders (2004), clustering is used to estimate the label probability for unlabelled data instances, which is the key component in measuring the utilities of each data instance. The approaches in this category argue that by training on the representative data instances, the classifier should be able to reach similar performance as training on the complete dataset.

In terms of the labelling cost, traditional active learning assumes a uniform cost for labelling data instances, which is argued to be an unrealistic assumption in real-world active

Algorithm 1 Cost-Sensitive Tree Sampling.

```
Initialize a tree structure with only a root node;
Assign all data instance to the root node;
while $total\_costs < Costbudget$ do
    //Query stage
    Leaf selection: Select a leaf node $v$ from the tree for querying;
    Instance selection: Select an unlabelled instance $z$ in $v$ to do the query;
    Query $z$'s label and cost;
    //Tree structure update stage
    Update the performance metric in the tree with $z$'s label and cost
    Tree reconstruction: find all nodes whose split needs renewal;
    Tree expansion: find all leaves that ready to be split;
    Update the tree structure by splitting the found nodes and leaves;
end while;
Label assignment: Construct the labelled dataset for base learner.
```

learning problems. Therefore, annotation cost-sensitive active learning is proposed to consider the real human annotation costs in the active learning algorithms.

In the paper, we focus on the general multi-class cost-sensitive active learning problems with single labeler and the costs remaining unknown before querying. There are various works targeting on annotation cost-sensitive active learning with different problem settings, such as the querying target (Greiner et al. 2002), the number of the labelers (Donmez and Carbonell 2008; Huang et al. 2017; Guillory and Bilmes 2009), the availability of annotation costs (Cuong and Xu 2016; Golovin and Krause 2011), the targeting classification problem (Yan and Huang 2018) and the applied data domain (Vijayanarasimhan and Grauman 2011; Liu et al. 2009). However, most of these works could not be intuitively applied to our problem setting due to the fundamental difference.

To discuss the cost-sensitive active learning with unknown costs, the question that ought to be answered first is whether the human annotation costs can be accurately estimated. In Arora et al. (2009) and Ringger et al. (2007), different unsupervised models are proposed to estimate the annotation costs for corpus datasets, while Settles et al. (2008) further shows that the annotation costs can be accurately estimated by using a supervised learning model.

In solving the cost-sensitive active learning problems, Tomanek and Hahn (2010) discusses the role of cost and the benefit (utility) in cost-sensitive active learning and proposed a querying strategy, return on investment, which combines the utility and cost in a measure. Haertel et al. (2008) compares three different querying strategies in cost-sensitive learning and demonstrates their performance with real-world datasets. However, the estimations of the utility and cost are usually taken as two independent tasks in cost-sensitive active learning, and the connection between them is lacking in discussion, leading to unreasonable settings in existing approaches. We will discuss this issue in the following section.

3 The proposed approach

The outline of CSTS is presented in Algorithm 1. There are two major stages in CSTS:

- *Query stage* Pick up the querying instance in the tree structure, which requires the selection of leaf and the selection of the querying instance within the leaf.
- *Tree structure update stage* Update the tree structure on the basis of the newly acquired label and cost, including renewing old split and splitting leaf nodes in the tree.

After the total costs meet the budget, a labelled dataset will be built by label assigning trick in order to train the base learner.

We shall discuss the detail of each stage of CSTS in this section. First, the motivation of the design of CSTS and the advantages in using tree structure to solve cost-sensitive active learning problems are demonstrated. Then, CSTS is proposed with the construction of tree-structured clusters, the query strategies and the label assigning trick explained in detail.

3.1 Background

The querying strategies in cost-sensitive active learning have two important components:

- **Utilities** the benefit that the classification task can gain for knowing the label of each data instance.
- **Costs** the prices we need to pay to acquire the labels of data instances.

The uncertainty is a popular criterion for estimating the utilities in cost-sensitive active learning algorithms (Haertel et al. 2008; Settles et al. 2008), yet a well-known drawback of the uncertainty sampling is the sample bias problem. The uncertainty measurement is highly related to the decision boundary in the trained classifier. Consider the case in Fig. 3 of Dasgupta 2011. If the initial boundary in the 1-D dataset is in the center group, the uncertainty sampling will be trapped in the decision boundary ω with 95% accuracy. However, the optimal boundary should be ω^*, which has 97.5% accuracy. On the basis of the example, if the data have multiple possible decision boundaries, the selected instances will be trapped in one of the boundaries on the basis of the initial queried data instances, resulting in the inconsistency of the algorithm or return a suboptimal solution (Dasgupta 2011). Aside from the model inconsistency issue, the sampling bias problem will also affect the cost estimate. In Settles et al. (2008), it was shown that a well-tuned regressor is capable of accurately estimating the annotation costs under 10-fold-cross-validation. However, the cross-validation setting in the experiments implies the demand on the unbiasedness of the training data. Therefore, when using bias samples from uncertainty-based cost-sensitive active learning algorithms to train the cost-estimating regressor, it is unlikely to yield promising result.

In the estimation of costs, unsupervised models are widely used in cost-sensitive active learning. For instance, Haertel et al. (2008) uses the hourly cost model (Ringger et al. 2007) to estimate the annotation costs of corpus datasets before applying active learning algorithms. In these approaches, the costs are estimated without knowledge of the labels. However, the labels may play an important role in accurately estimating the costs. Consider the case in which doctors attempt to diagnose a disease in a group of patients; doctors will likely need more time to arrive at a diagnosis for patients without obvious symptoms. In the example, the annotation costs are highly related to the decision boundary in the label prediction; therefore they cannot be accurately estimated without considering the information from the labels. Furthermore, these unsupervised models can be only applied to restricted data types, as the hourly cost model only works for corpus datasets. Owing to the issue, cost-sensitive active learning algorithms that rely on unsupervised cost-estimation models are constrained to limited applications, losing their versatility.

To avoid the problems mentioned above, our proposed method aims to estimate the cost and utility jointly with the tree-structure clusters. There are many advantages in using tree-structured clusters for active learning:

- *Solve the model inconsistency issue in the sample bias problem* (Dasgupta and Hsu 2008). Moreover, the sampling bias will no longer affect the cost estimate as mentioned

above, owing to the independence between clusters and the uniform estimation of all data instances within the same cluster.
- *Give feasibility when modeling the label and cost distributions.* The tree structure allows us to reconstruct an impure cluster without reclustering and affecting other clusters.
- *Mimic the behavior of uncertainty sampling.* During the exploration of tree-structured clusters, the clusters close to the decision boundary will suffer from a low label purity; hence, more labels are required to build smaller clusters when replacing them in order to better model the label distribution. As a consequence, the active learning algorithm will favor querying data instances close to the boundary, which follows the querying strategy in uncertainty sampling.

We then extend the usage of tree-structured clusters to the cost-sensitive active learning. With the assumption that neighbors share similar labels and costs, we could further argue that with a set of fine-grained clusters, data instances in the same cluster should share the same label and similar costs. By estimating the utility with the label uncertainty and using the average known costs as the estimated cost, we could gradually find the major label in each cluster as the amount of queried data increases, and further label the entire dataset if the label purity permits.

In real world datasets, the assumption of data instances having similar costs with neighbors may not perfectly hold. In the case that datasets come with drifts between the costs of certain pairs of neighboring instances, our tree-structured clusters shall remain effective as the drift can be handled by splitting these pairs into different nodes. However, if the dataset completely break the assumption, i.e., the costs in the dataset are no longer continuous, then the costs prediction becomes an unsolvable problem for all approaches if no further information is provided.

The practicality of the main assumption of clustering-based active learning algorithms, *data instances belong to the same cluster are likely to share the same label*, remains a problem. Since clustering algorithms are unsupervised approaches, label information is assume unknown or ignored in the algorithms. Therefore, there is no guarantee on the label purity in clusters, making the assumption in clustering-based active learning algorithms questionable, especially on the real-world datasets. This is also a seriously problem preventing clustering-based active learning algorithms from practical usage.

Dasgupta and Hsu (2008) proposed an innovative approach to find out high purity clusters. To solve a traditional pool-based active learning problem, they start by utilizing the hierarchical clustering algorithm to model data instances into tree-structured clusters. An active learning strategy is then proposed to discover and exploit informative pruning of the cluster tree. The work mainly focuses on solving the model inconsistency issue in the sample bias problem. On the other hand, the hierarchical clustering structure guarantee the existence of pure label clusters within the tree structure. However, The algorithm still requires the existence of high purity clusters in the top layers of hierarchical structure, leading to poor performances in real world datasets that fail to meet the requirement. Nevertheless, it still reduces the impractical assumption problem, and gives an inspiration on our proposed method.

While tree-structured clusters is built in advance under a bottom-up hierarchical clustering approach in Dasgupta and Hsu (2008), it is hard to integrate label and cost information within the cluster structure. Therefore, we design a revised decision tree algorithm which can build the tree-structured clusters in a top-down manner during the query stage. The top-down building approach allows us to integrate known cost and label information, which can be regarded as a supervised approach, leading to a better guarantee on label purity and cost similarity than other unsupervised approaches. Combining with a designed query strategy, the cost-sensitive tree sampling (`CSTS`) is then proposed to solve the cost-sensitive active learning problems, which can divided into three parts: tree-structured clusters construction, queried instance selection and label assignment.

3.2 Tree-structured clusters construction

Here, we shall discuss the revised decision tree algorithm for constructing the tree-structured clusters. The original decision tree algorithm is first discussed. Then, we propose a novel metric to evaluate the quality of a node, following with the detail design in the tree-structured clusters construction.

3.2.1 Decision tree algorithm

In machine learning, the decision tree algorithm is a well-known tree-structured model (Quinlan 1986, 2014; Breiman et al. 1984). The procedure of training a decision tree model includes three important components: a splitting method, an evaluation metric, and a stopping criterion.

Splitting method Node splitting is mostly carried out by a simple decision stump algorithm, which uses a simple threshold for a specific dimension of a feature to split data instances into two halves. This part remains the same in our proposed algorithm.

Evaluation metric Although there are many possible ways to split a node, an evaluation metric is needed to define which one is the best. In the cost-sensitive active learning problem, the tree-structured clusters should be able to model both the label and cost; therefore, a novel evaluation metric is proposed to fulfill this goal.

Stopping criterion The stopping criterion is key for preventing the model from overfitting. In a binary classification problem, consider the following VC inequality (Vapnik 2013) for the nodes of the decision tree model:

$$\mathbf{P}\left[\sup_{\mathbf{f}\in\boldsymbol{F}}\left|\hat{R}_N\left(\mathbf{f}\right)-R\left(\mathbf{f}\right)\right|>\epsilon\right]\leq 8S(\boldsymbol{F},N)\exp\left(-N\epsilon^2/32\right),$$

where $\boldsymbol{F}$ is the collection of all possible decision stump classifiers, $\hat{R}_N\left(\mathbf{f}\right)$ is the 0/1 loss of classifier $\mathbf{f}$ for N training instances, $R\left(\mathbf{f}\right)$ stands for the testing error for classifier $\mathbf{f}$, and $S(\boldsymbol{F}, N)$ is the growth function of decision stump, which is $2N$. The inequality says that the upper bound of the probability for the deviation between the training and testing errors being less than ϵ is proportional to $16N\exp(-N\epsilon^2/32)$, which is a monotonically decreasing function for $N \geq 1$. Therefore, for a small N, the algorithm may overfit on the labelled data, leading to a difference between training error and testing error. We can extend the discussion to the worst case of our active learning algorithm, i.e., querying instances randomly. The issue becomes even harsher owing to the limited amount of labelled data. As a result, how to prevent our tree-structured clusters from overfitting on the labelled datasets is an essential problem to tackle. Here, we delicately solve the issue within the evaluation metric, which will be discussed in detail in the following section.

Table 1 Key quantities in the algorithm

L	Total number of classes in the dataset
n_v	Number of labelled data instances in node v
s_v	Number of data instances in node v
$f_{v,l}$	Real label l fraction in node v
$\hat{f}_{v,l}$	Known label l fraction in node v from labelled data
$\tilde{f}_{v,l}$	Estimated label l fraction in node v
C_v	The set of known costs in node v

3.2.2 Metric for node evaluation

Owing to the demands of the evaluations on cost similarity and label purity, we propose a novel metric to measure the quality of tree nodes, which contains two parts: the Gini impurity and cost variance.

Gini impurity The Gini impurity is a famous metric that is used by the CART algorithm (Breiman et al. 1984) (Table 1). When solving an L-classes classification problem, let $f_{v,l}$ be the fraction with label l within the node v. Then, the Gini impurity is computed by

$$G(f_v) = \sum_{l=1}^{L} f_{v,l}(1 - f_{v,l}).$$

However, in an active learning setting, the number of labelled instances inside the node may be too few to approximate the true label fraction. To solve this issue, we design an inherit approach to estimate the label fraction for all nodes except the root.

Inside node v, s_v denotes the total number of data instances, n_v is the number of labelled instances, and $\hat{f}_{v,l}$ stands for the fraction with label l in the n_v labelled instances. By assuming data instances within a node are i.i.d., the estimation of the true label fraction $f_{v,l}$ is actually the same as the estimation of a Bernoulli distribution by conducting n_v i.i.d experiments if we regard the appearance of the label l as an outcome 1 and the rest as 0. Therefore, we can use the length of the normal approximation confidence interval to show how confident we are using $\hat{f}_{v,l}$ to estimate $f_{v,l}$:

$$\Delta_{v,l} = \sqrt{\frac{1}{n_v}\hat{f}_{v,l}(1 - \hat{f}_{v,l})}.$$

In contrast to the traditional distribution approximation problem, we have a limited number of samples s_v, which means as n_v approaches s_v, the length of the confidence interval should become smaller. Moreover, for computational convenience, we use a global confidence interval for all of the labels within the same node. On the basis of these facts, we propose the following revised confidence interval:

$$\hat{\Delta}_v = \left(1 - \frac{n_v}{s_v}\right) \times \frac{1}{n_v} + \left(1 - \frac{n_v}{s_v}\right) \times \left(\prod_{l=1}^{L} \sqrt{\frac{1}{n_v}\hat{f}_{v,l}(1 - \hat{f}_{v,l})}\right)^{\frac{1}{L}}. \qquad (1)$$

The $1 - \frac{n_v}{s_v}$ term considers the situation with limited samples; $\frac{1}{n_v}$ ensures that $\hat{\Delta}_v \neq 0$ when only a single type of label is queried and that the confidence interval will become larger if less labelled data exist, which prevent the tree from over-expanding; the last term is the geometric mean of $\Delta_{v,l}$ for all labels.

On the basis of revised confidence interval, we define the estimated label fraction as

$$\tilde{f}_{v,l} = \begin{cases} \infty & \text{if } \alpha \hat{\Delta}_v \geq 1 \\ (1 - \alpha \hat{\Delta}_v) \hat{f}_{v,l} + \alpha \hat{\Delta}_v \tilde{f}_{p,l} & \text{else} \end{cases}, \tag{2}$$

where d is the depth of node v, and $\tilde{f}_{p,l}$ is the estimated label fraction of its parents (set as uniform label fractions for the top node). α is a parameter that can control how much we trust in $\hat{f}_{v,l}$. To simplify the algorithm, we set $\alpha = 2$ in our experiments.

The design of the estimated label fraction has two key points:

- **Preventing overfitting** As we mentioned in previous section, how to prevent overfitting is an important question that ought to be answered in building the tree-structured clusters. In our algorithm, the combination of parameter α and the revised confidence interval $\hat{\Delta}_v$ can fulfill the target. A smaller value of $\hat{\Delta}_v$ indicates the higher confidence that the known label fractions $\hat{f}_v$ are close to the true label fractions f_v, which is less likely for the node v to be overfit. Combine with a parameter α to feasibly control the threshold, a judgment $\alpha \hat{\Delta}_v \geq 1$ is proposed to define when we are confident enough to build node v without overfitting. If a node meet the judgment, it indicates the revised confidence interval is too large to make current estimation trustworthy. As a results, infinite values are assigned to the label fractions, which further prevent the parent node from being split into the case.
- **Conservative inheriting estimation** While the revised confidence interval indicates the trust level of the known label fractions, it needs to be integrated into the estimated label fractions to better decide the optimal split of a node. In our designed approach, we use the formula of inner division point to calculate estimated label fractions $\tilde{f}_v$, the weights of the known label fractions are decided by $1 - \alpha \hat{\Delta}_v$, which depends on the length of confidence interval, showing how trustworthy the known label fractions are. The remain part inherits the estimated label fractions $\tilde{f}_p$ from the parent p. The inheriting fraction stands for the result of the split overfit on the known labels, so a part of the unknown labels still follow the parent's fractions. The design yields a conservative estimation on the label fractions, which also prevents the algorithm to over-trust the partial label information.

In summary, Eq. (2) prevents our tree model from overfitting, and utilizes the partially inheriting approach to integrate the confidence information into the estimated label fractions.

On the basis of Eq. (2) and the original Gini impurity formula, the Gini impurity of node v is estimated by

$$G_e(\tilde{f}_v) = \sum_{l=1}^{L} |\tilde{f}_{v,l}(1 - \tilde{f}_{v,l})|. \tag{3}$$

Cost variance In order to carry out cost estimation, the cost variance is another factor in the node evaluation, which is calculated by

$$V(C_v) = \frac{1}{n_v^2} \sum_{i=1}^{n_v} \sum_{j=1}^{n_v} \frac{1}{2} (C_{v,i} - C_{v,j})^2$$

with the known annotation costs C_v for n_v labelled data instances.

Scaling of the annotation costs is required when constructing the performance metric. Since the optimal value for both the Gini impurity and cost variance is 0, the main idea is to scale the worst case of the cost variance to the same value as the Gini impurity, which is computed by

$$V_{scaled}(C_v) = V(C_v) \times \frac{G_w}{V_w}, \tag{4}$$

where

$$\begin{cases} G_w &= \sum_{i=1}^{L} \frac{1}{L}(1 - \frac{1}{L}) = 1 - \frac{1}{L}, \\ V_w &= \frac{1}{4}(max(C_p) - min(C_p))^2, \end{cases}$$

and C_p is the annotation costs of v's parent.

The largest value of the Gini impurity for an L-class classification problem can be computed by the Cauchy-Schwarz inequality. As to the worst case of the cost variance, we use the largest variance for all possible splits from the node, which is splitting only two labelled data instances with the largest and smallest costs to the same node.

Performance metric Combining both factors, the metric used for the evaluation is defined as

$$M(v) = (1 - \beta) \times G_e(\tilde{f_v}) + \beta \times V_{scaled}(C_v), \tag{5}$$

where $G_e(\tilde{f_v})$ and $V_{scaled}(C_v)$ are defined in Eqs. (3) and (4), and β is a parameter for adjusting the importance of these two factors. For simplicity, we use $\beta = 0.5$ in our experiments, which means that both factors have the same weights in the evaluation.

3.2.3 Tree structure construction

Our model starts from all of the data instances in the root node. As more and more labels are queried, the tree will be expanded until no further split on leaves can improve the performance metric. Furthermore, we also propose a reconstruction criterion to ensure all expansions are actually improving the performance metric during the query stage.

Tree expansion The procedure of tree expansion is as follows: Use the decision stump algorithm on each data dimension to determine a way to split the node that leads to the optimal performance metric. If the expansion does help in improving the performance, we say that the node is ready to be split. To emphasize, the confidence interval we propose in Eq. (1) takes the amount of labelled data instances and the label distribution in to consideration. Therefore, when combining it with parameter α in Eq. (2), the user can adjust the tolerance on the confidence level to prevent the tree model from overfitting.

Tree reconstruction Even though we can prevent a tree from overfitting, we cannot guarantee that all expansions are effective at any time. In the other words, we may find that some splits of nodes in the tree do not optimize the value of the performance metric when we acquire new labels and costs. Therefore, during the query stage, the model will keep track of the metric values of all nodes. Once it finds a split of a node that leads to a poorer metric value, it will mark the node as dirty, so the split of the node and its subtree will be renewed during the tree structure update.

Tree structure update In order to reduce the computational overhead and improve the regularization, we use a dirty bit to indicate if the node is ready to be split or needs reconstruction. Notice that if a node needs reconstruction, all of the nodes in its subtree will be set as dirty. Once the proportion of dirty leaves exceeds a certain threshold γ, our algorithm will update the tree structure from the top of the tree by finding the best split of the dirty nodes and expand them until no leaf node can be expanded. The detail of tree structure update is showed in in Algorithm 2.

3.3 Cost-sensitive tree sampling

In using the proposed tree-structured clusters to solve the cost-sensitive active learning problems, how to pick the querying instance is another key issue. In the following, we shall discuss the two important components in the query stage: leaf selection and instance selection. After that, the label assigning trick are introduced, which could further improve the performance of CSTS. The full design of CSTS is listed in Algorithm 2.

3.3.1 Query stage

To use the tree model to solve cost-sensitive active learning problems, it is necessary to specify the selection of data instances to query their labels. During the query stage, the instances are always selected from a leaf inside the tree structure; therefore, the problem can be split into two parts: leaf selection and instance selection within leaves.

Leaf selection The estimated impurity multiplied by the number of unlabelled data instances within a node stands for the estimate of the utilities in our proposed approach, whereas the estimated cost is the average cost of a labelled data instance in the cluster. Combining both factors, the probability that a leaf node v is selected is as follows:

$$p_v \propto \underbrace{G_e(\tilde{f}_v) \times (s_v - n_v)}_{\text{utility}} / \underbrace{\bar{C}_v}_{\text{cost}}, \tag{6}$$

where $\bar{C}_v$ is the average annotation cost in the leaf node.

Some analysis of the behavior of the query criterion. The estimated utility includes the estimated impurity and the number of unlabelled data instances. The estimated impurity is calculated by Eq. (3), which considers both the estimated confidence and label impurity issue. Therefore, the proposed query criteria is a coordination of four elements in a leaf: estimated confidence, label impurity, number of unlabelled instances and estimated costs. Since the effect of the estimated costs on the query criterion is straightforward, we focus only on the behavior of utility and assume an uniform cost for simplicity.

Consider a case that two children from the same parent share the same quantities, except that the first child has less labelled data instances than the second one. Therefore, the first child have a longer confidence interval, leading to a larger inherit fraction and end up with higher estimated Gini impurity. As a results, we could observe the favor to enhance the less confident node in our designed query criteria. On the other hands, suppose two children have different label fractions while other quantities remain the same, the query criteria will naturally focus on the one with higher Gini impurity value and select more instances to query from it.

On the other hand, considering the number of unlabelled data instances in our query criteria can balance the query selection. Suppose there are only two labels, and only two same sized clusters are found. Cluster A has 5% of minority label, and can be split out by decision stump, while cluster B has even label fractions (50% each) and cannot be improved by further splitting. Our query criteria will first focus on improving cluster B, even though the impurity cannot be reduced after queries in the cluster B. As the number of unlabelled

Algorithm 2 Cost-Sensitive Tree Sampling.

```
Input: pool of unlabelled data $P$; Cost budget $B$
$T \leftarrow \{root\}$; (all data instances in the root)
while $current\_budget < B$ do
    while $dirty\_leaf\_proportion < \gamma$ do
        //Query stage
        Leaf selection: $v \leftarrow$ select_leaf($T$) base on (6);
        Instance selection:$z \leftarrow$ select_instance($v$, P);
        Query $z$'s label and cost;
        remove $z$ from $P$;
        Update performance metric for all nodes in $T$;
        Tree reconstruction: Set nodes that need split renewal as dirty;
        Tree expansion: Set leaves that ready to be split as dirty;
    end while;
    //Tree structure update
    for all dirty nodes $n \in T$ in top-down order do
        set $n$ as not dirty;
        $T \leftarrow T \setminus$ {subtree of $n$};
        $children \leftarrow decision\_stump(n)$;
        if $children \neq Null$ then
            set $children$ as dirty;
            $T \leftarrow T \cup children$;
        end if
    end for
end while;
for all leaf node $v \in T$ do
    if $G_e(\tilde{f_v}) < t$ then
        Label assigning: Assign majority label to all unlabelled data in $v$
    end if
end for
```

data instance gradually drops in cluster B, the query criteria may turn to select data instance from cluster A, and therefore, take the chance to further improve the label purity.

To sum up, the designed query criteria considers the trade off between costs and utilities, where includes label purity, label confidence and label instances balancing, making it an strong query criteria in solving cost-sensitive active learning problems with our proposed tree-structured clusters.

Instances selection within leaves We use the length of the confidence interval $\hat{\Delta}_v$ to measure whether a node is trustworthy with its labelled data. Therefore, regardless of the method for selecting queried instances, it will not change the guarantee of the stable condition in the performance metric, which gives us flexibility in choosing data instances inside a leaf. The followings are some options:

- **Random sampling** Choose any unlabelled instances with uniform probability.
- **Uncertainty sampling on a classifier** Choose the instance with the minimum distance to current decision boundary or tge maximum value of any uncertainty measurement.
- **Representative sampling** Choose the unlabelled data instance closest to the centroid of the selected leaf.
- **Least cost sampling** Choose the unlabelled data instance closest to instance with the lowest cost in the selected leaf.

Here, we simply use the random sampling approach in all the experiments, which choose any unlabelled instances to query with a uniform probability.

3.3.2 Label assigning trick

At the end of the query stage, a natural ability of the tree model is to label all the unlabelled data instances with the majority label inside every leaf node. However, as the evaluation metric for tree construction does not focus entirely on enhancing the label purity, it is unlikely to accurately label all the data instances. Therefore, if we simply label all the data instances, it may made a large number of mistakes at those impure nodes and further degrade the performance of the model trained on it. A design in our model is that the estimated Gini impurity can properly reflect the label purity within each nodes. Hence, we introduce another parameter t as a threshold on the estimated Gini impurity, and only regard the majority label in a cluster as trustworthy when the impurity $\leq t$ and assign the majority label to the unlabelled data instances.

4 Experiments

In the section, we shall first discuss the experiment settings, followed by quantitative comparisons between CSTS and other state-of-the-art competitors on three types of datasets: dataset with artificial costs, dataset with attribute costs and dataset with real annotation costs. Finally, extensive experiments on CSTS using different parameters are conducted in order to analyze the parameter sensitivity.

4.1 Experiment setting

In all the following experiments, we simply set parameters $(\alpha, \beta, \gamma) = (2, 0.5, 0.5)$ in CSTS to generalize the model, despite the fact that further experiments show the performance of CSTS can be improved if an optimal set of parameters is used. Parameter t is tuned with 4-fold-cross-validation by adding self-labelled data instances to the training validation set for different values of t in order to coordinate the various properties of the datasets.

We compared CSTS with four different methods: random sampling (RS), cost-sensitive hierarchical sampling (CSHS), return on investment (ROI) (Haertel et al. 2008) and rank combination cost-constrained sampling (LRK) (Tomanek and Hahn 2010).

- RS: Random sampling is a baseline approach that chooses a data instance to label at random, ignoring both the utility and the annotation cost.
- CSHS: Cost-sensitive hierarchical sampling is a transformation of hierarchical sampling (Dasgupta and Hsu 2008). The only change is to divide the original probability for choosing a leaf to query by the average of known annotation costs in the leaf, turning it into a cost-sensitive method.
- ROI and LRK: These methods are two different ways of combining the utility and cost to form the query criteria. To estimate the utilities, both of them use the uncertainty measurement, calculated by the entropy of the predicted probabilities in each class. The costs are estimated by a regression tree with tuned limited depth $\in [5, 10, 15, 20, 25]$, in order to give even comparisons with our tree-structured model. Note that the tree model will be retrained every time when a data instance is newly queried and added to the training set. In ROI, it uses the ratio of the utility and cost as the selection criterion; LRK determines the ranks of data instances for the utility and cost and combines the rank with a ratio, which we set as 0.5 in our experiments, to form the metric for choosing data instances to label.

Table 2 Characteristics of UCI datasets

Dataset	N	L	Dataset	N	L	Dataset	N	L
liver	345	2	*german*	996	2	*knowledge*	258	4
vote	411	2	*mushroom*	8124	2	*vehicle*	846	4
breast	699	2	*adult*	32, 561	2	*nursey*	12, 960	5
diabetes	764	2	*seeds*	210	3	*yeast*	1484	9

We used a l_2-regularized logistic regression as the base learner to be trained on the label data we acquire from active learning for all methods. The parameters of the base learner are tuned by 4-fold-cross-validation independently in each method. For all datasets, we reserve 80% of the data as the data pool and retain 20% as the final testing set. The presented results are the average over 10 times of experiments, and the budget of label costs grows in small increments with a maximum 30% of the total costs.

4.2 Datasets with artificial costs

4.2.1 Dataset

We compare CSTS with other competitors on twelve datasets from the UCI Repository (Lichman 2013) with artificially created annotation costs. The size of the datasets N and the number of classes L are summarized in Table 2.

4.2.2 Artificial costs creation

The annotation cost is created on the basis of two assumptions:

- *The data instances closed to the decision boundary should have larger costs.*
- *The cost distribution should have a connection to the data distribution.*

The first assumption is based on the argument that if a data instance is closer to the decision boundary, the feature that we can directly use to classify it is less clear, so the oracle (usually human beings) will need to spend more effort to correctly label the data instance. On the other hand, two similar data instances should have similar annotation costs, indicating that the data distribution is essential in creating the annotation cost, which implies the second assumption.

Figure 1 demonstrates the procedure to create the costs. Notice that the cost creation is independent from the active learning, so we assume all the labels in datasets are available.

1. Utilize a SVM model with RBF kernel and the parameter $C = 100$ to fit on the datasets, take the hyper-planes as the decision boundaries in the oracle.
2. Model the data distribution, where the k-means clustering with $k = \mathtt{N}/10$ is used. Base on the assumption (2), we simply assume that those data instances in the same cluster share the same annotation cost.
3. Calculate $\bar{D}_v$, the average distance to the closest decision boundary for data instances in v, which is used to construct the *reverse distance cost* and the *distance cost*.

The *reverse distance cost* takes the original assumption by setting the annotation cost of data instances that belong to cluster v as $1/\bar{D}_v$. However, the setting will lead to a dilemma

Fig. 1 Procedure of cost creation

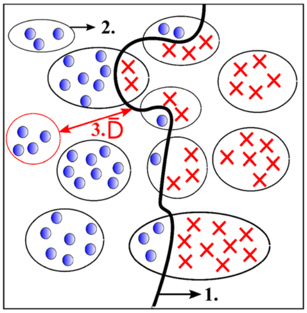


Table 3 AUC scores for UCI datasets with the reverse distance cost

Datasets	CSTS	RS	CSHS	ROI	LRK
liver	16.58/1.749(4)	16.98/2.257(3)	14.99/1.573(5)	**18.29/0.782(1)**	17.63/1.341(2)
vote	26.41/0.888(2)	**26.46/0.783(1)**	25.66/1.182(5)	26.3/0.49(4)	26.33/0.65(3)
breast	**27.47/0.485(1)**	27.42/0.481(2)	27.23/0.371(3)	27.06/0.362(4)	27.01/0.4(5)
diabetes	22.29/1.09(3)	22.14/0.709(4)	21.32/1.165(5)	**22.65/0.499(1)**	22.59/0.552(2)
german	**21.03/0.681(1)**	20.66/0.865(2)	20.38/1.083(3)	20.14/0.424(5)	20.37/0.523(4)
mushroom	**27.32/0.449(1)**	27.21/0.482(3)	27.27/0.223(2)	26.97/0.141(5)	27.0/0.155(4)
adult	**23.97/0.066(1)**	23.92/0.086(2)	23.66/0.08(5)	23.89/0.037(4)	23.9/0.031(3)
seeds	**25.68/1.274(1)**	25.03/1.333(4)	23.34/2.575(5)	25.54/1.078(2)	25.24/1.241(3)
knowledge	**19.65/1.866(1)**	19.36/2.712(3)	15.83/2.232(5)	19.36/1.093(2)	19.16/1.217(4)
vehicle	**19.85/0.825(1)**	19.29/1.443(2)	14.91/1.1(5)	19.11/0.638(4)	19.22/0.539(3)
nursey	21.79/0.175(3)	21.71/0.176(4)	19.88/0.582(5)	21.86/0.103(2)	**21.86/0.091(1)**
yeast	**15.64/0.639(1)**	15.16/0.61(2)	14.52/1.081(5)	14.93/0.345(4)	15.03/0.341(3)
sum_of_ranks	20	32	53	38	37

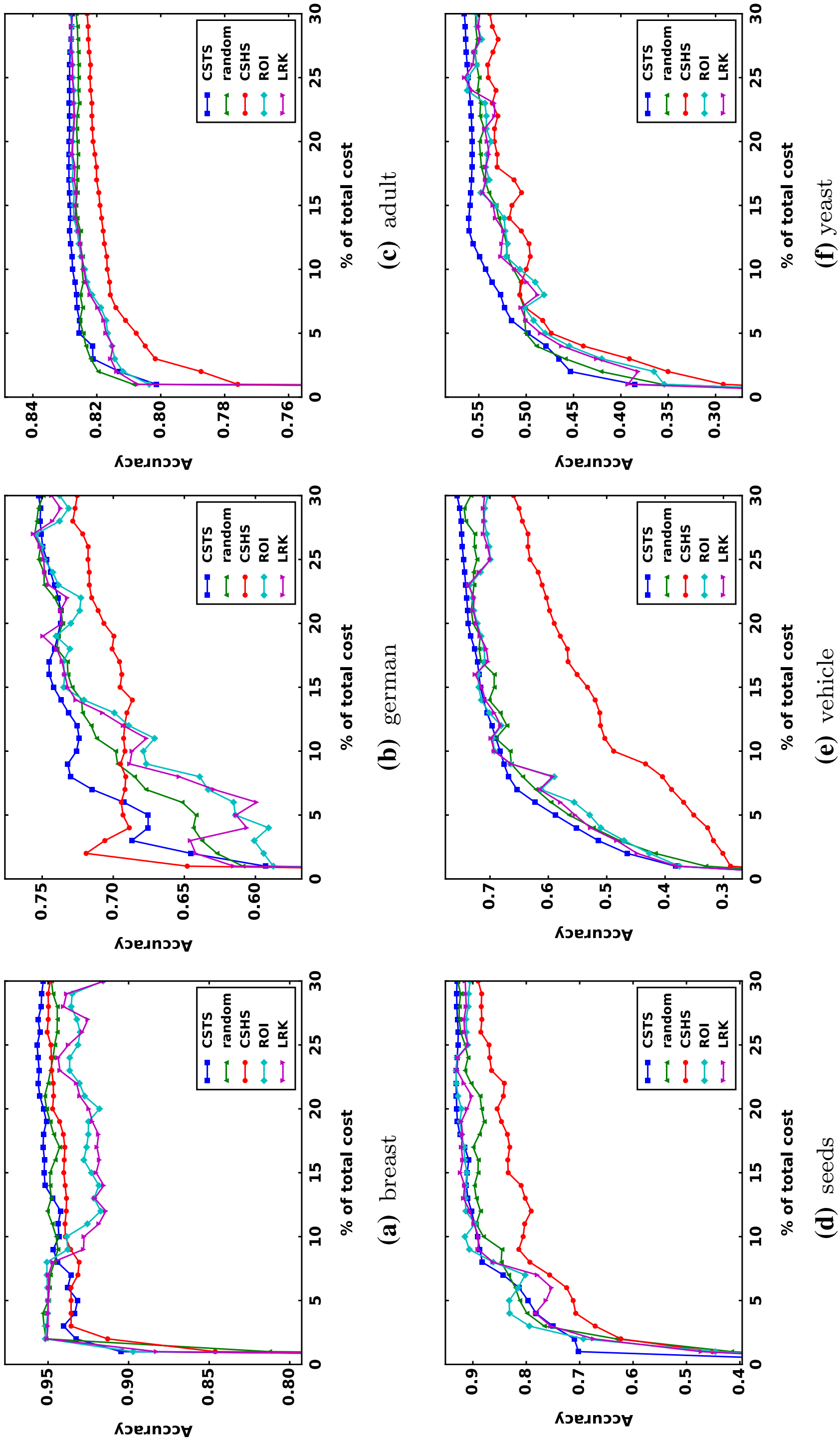
The toppest rank entry is marked in bold

in uncertainty sampling since the instances with higher costs are also the ones that are most informative in their criteria. Therefore, we also conduct experiments on *distance cost*, setting the average distances to the boundaries $\bar{D}_v$ in cluster v as the annotation cost. The setting gives a considerable advantage to uncertainty-sampling-based algorithm, in order to observe if CSTS is able to adapt to different cost settings and provide comparable performance.

4.2.3 Experiments results on *reverse distance cost*

Table 3 shows the area under the cost/accuracy curve (AUC) of algorithms for all twelve datasets under *reverse distance cost*. Figure 2 shows the results for six selected datasets. We also compare the mean accuracy scores from different algorithms when setting different percentages of the total cost as the budget. Table 4 summarizes a comparison of CSTS against the others using a two-sample t-test at 95% significance, which gives better insights into the model performance at each query stage.

The results in Table 3 clearly indicates that none of the algorithms are capable of providing consistently superior performance. Despite this, CSTS has the best performance for 8 datasets, which is the most compared with the others. Regarding the overall performance, the sum of the ranks indicates that CSTS is still the best model on average.



(a) breast **(b)** german **(c)** adult **(d)** seeds **(e)** vehicle **(f)** yeast

Fig. 2 Test accuracy for selected UCI datasets

Table 4 Win/tie/lose based on a two-sample t-test

Base: CSTS	Percentage of total queried instance's costs						Total
Competitors	5%	10%	15%	20%	25%	30%	
RS	2/9/1	2/10/0	2/10/0	2/10/0	1/11/0	1/11/0	10/61/1
CSHS	5/7/0	8/4/0	11/1/0	9/3/0	10/2/0	7/5/0	50/22/0
ROI	5/5/2	4/6/2	2/8/2	6/4/2	4/7/1	3/7/2	24/37/11
LRK	5/5/2	4/6/2	2/8/2	5/6/1	4/5/3	3/8/1	23/38/11
total	17/26/5	18/26/4	17/27/4	22/23/3	19/25/4	14/31/3	107/158/23

There are two important observation in the table:

- Our proposed approach significantly outperforms CSHS. The observation indicates that the supervised tree-structured clusters building approach can lead to a better performance than traditional unsupervised approach.
- Our proposed approach have better performance in most of the datasets in comparison with ROI and LRK, proving that CSTS can be a better approach in solving cost-sensitive active learning problems than uncertainty-based methods.

We can also observe that CSHS, ROI and LRK have worse performance than the random sampling in parts of the datasets. As discussed in Sect. 3, a main criticism for the clustering based active learning approaches is that they over-rely on the performance of unsupervised clustering methods. Therefore, the CSHS can only provide effective performance on few datasets (*german* and *mushroom*) which can be easily modelled by the clusters. The uncertainty based approaches, ROI and LRK, suffer from the contradiction between the cost setting and their query strategy. Besides, the cost learning in these approaches are based on the biased data, making it harder to give accurate predictions on the annotation costs. Our approach, on the other hand, can deal with all the issues mentioned above and yield promising performance.

From the two-sample t-test comparison in Table 4, we can observe that CSTS is mostly able to provide comparable or even better accuracy than the competitors in each stage, especially when the budget reach around 20% of total costs. While the early stage are highly affected by the initially queried instance and all models have sufficient labelled instances to provide similar performance in the later stage, the superiority of CSTS in the middle stage strongly prove its effectiveness once again.

4.2.4 Experiment results on distance cost

Table 5 present the experiment results on *distance cost*. Compared with the *reverse distance cost*, both LRK and ROI have a slight improvement on the sum of ranks and also have the best AUC for more datasets; in the mean time, the ranks of CSTS slightly drop under the new setting. However, comparing the sum of the ranks, CSTS still provides the best performance under such a favorable setting for uncertainty sampling, indicating that our proposed method is able to handle different cost distributions and yield stable performance.

Table 5 AUC scores for UCI datasets with the *distance cost*

Datasets	CSTS	RS	CSHS	ROI	LRK
liver	16.92/2.072(4)	16.93/1.779(3)	14.13/1.429(5)	17.67/1.575(2)	**17.84/1.323(1)**
vote	**26.55/0.932(1)**	26.39/0.771(3)	25.55/1.136(5)	26.33/0.53(4)	26.41/0.506(2)
breast	**27.48/0.381(1)**	27.4/0.446(2)	27.33/0.314(3)	26.89/0.205(5)	27.0/0.311(4)
diabetes	22.41/0.826(3)	22.36/0.942(4)	21.47/0.741(5)	**22.56/0.508(1)**	22.46/0.467(2)
german	**21.16/0.781(1)**	20.66/0.694(2)	20.48/1.231(3)	20.28/0.441(4)	20.24/0.501(5)
mushroom	**27.34/0.503(1)**	27.18/0.319(3)	27.31/0.226(2)	26.97/0.154(5)	26.99/0.129(4)
adult	**23.96/0.08(1)**	23.92/0.065(2)	23.7/0.077(5)	23.88/0.037(4)	23.88/0.031(3)
seeds	25.37/1.582(3)	25.06/1.286(4)	23.88/1.907(5)	25.37/1.289(2)	**25.38/1.103(1)**
knowledge	19.05/2.423(4)	19.34/2.066(3)	15.16/2.777(5)	**19.46/1.337(1)**	19.38/1.021(2)
vehicle	**20.04/1.01(1)**	19.56/1.362(3)	14.58/1.433(5)	19.61/0.475(2)	19.34/0.591(4)
nursey	21.71/0.266(3)	21.7/0.143(4)	19.84/0.525(5)	21.87/0.096(2)	**21.88/0.101(1)**
yeast	15.28/0.712(2)	**15.52/0.626(1)**	14.46/0.941(5)	14.77/0.448(4)	14.86/0.423(3)
sum_of_ranks	25	34	53	36	32

The toppest rank entry is marked in bold

4.3 Dataset with attribute costs

4.3.1 Dataset

The UCI (Lichman 2013) **german** dataset collects attributes for a group of people to predict their credit risk. Among the attributes, one of them is the duration of people's checking account in month. As a longer duration of the checking account requires longer time to analyze, the cost for human beings to label an data instance should be highly related to the duration of the checking account. Therefore, we remove the duration attribute from the dataset and use it as the costs in our ACSAL setup, in order to observe if CSTS could adapt to the nearly real world costs setting.

4.3.2 Experiment results

Figure 3 shows the test accuracy. We can observe that CSTS stably outperforms the other approaches. In the mean time, CSHS yields a comparable performance in the early stage. The experiments results show the superiority of clustering based active learning approach, indicating that the dataset can be easily modeled by the clusters. On the other hand, both the uncertainty based approach, ROI and LRK, only have similar performance as random sampling. In summary, the experimental results shows the superiority of CSTS and the effectiveness of clustering based active learning approaches.

4.4 Dataset with real annotation costs

4.4.1 Dataset

Due to the limited number of datasets with annotation costs information, we only conduct experiments on one dataset, the *Speculative Text Corpus* dataset provided by Settles et al.

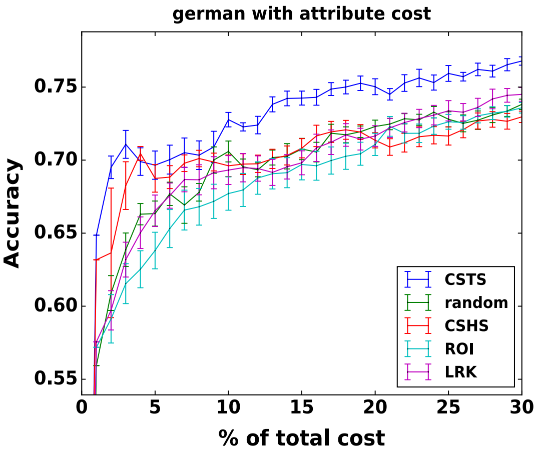


Fig. 3 Test accuracy of CSTS and the competitors

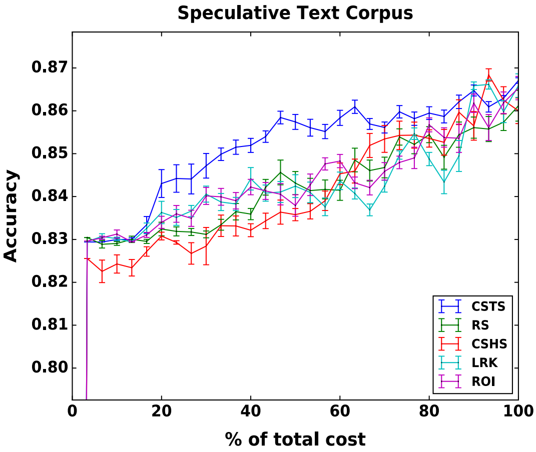


Fig. 4 Test accuracy of CSTS and the competitors

2008, to show the performance of CSTS for real annotation costs. It includes 850 sentences labelled to be speculative or definite by three different people, and the average annotation time is taken as the cost. In the experiments, we use a bag-of-words representation for the sentences, removing stopping words and terms with a frequency less than two.

4.4.2 Experiment results

Figure 4 shows the test accuracy. We can observe that CSTS outperforms the other approaches as the budget increases to 10% of the total cost. The trend is consistent with the design: In the early stage, the number of labelled data instance may be too few to build high quality clusters, which leads to the similar performance with random sampling; as more and more labels and costs we obtain, high quality clusters can be built and a significant improvement on the model performance can be observed. In summary, the experimental results indicate that our proposed approach is capable of providing superior results in comparison with four other methods for the dataset with real annotation costs.

4.5 Parameter sensitivity analysis

4.5.1 Parameters

We further conduct experiments to analyze the parameter's impact on the model performance. There are four important parameters in our proposed methods: In Eq. 2, α controls the proportion of known label fraction $\hat{f}$ in estimated label fraction $\tilde{f}$ and the threshold on confidence interval in deciding whether the estimation is trustworthy; β decides the balance between Gini impurity and cost variance in the evaluation metric; γ is the threshold on the fraction of dirty leaves, controlling when the tree structure should be updated; t decides whether the unlabelled data instances in a cluster can be labelled with the majority label and be included in the training dataset for the base learner.

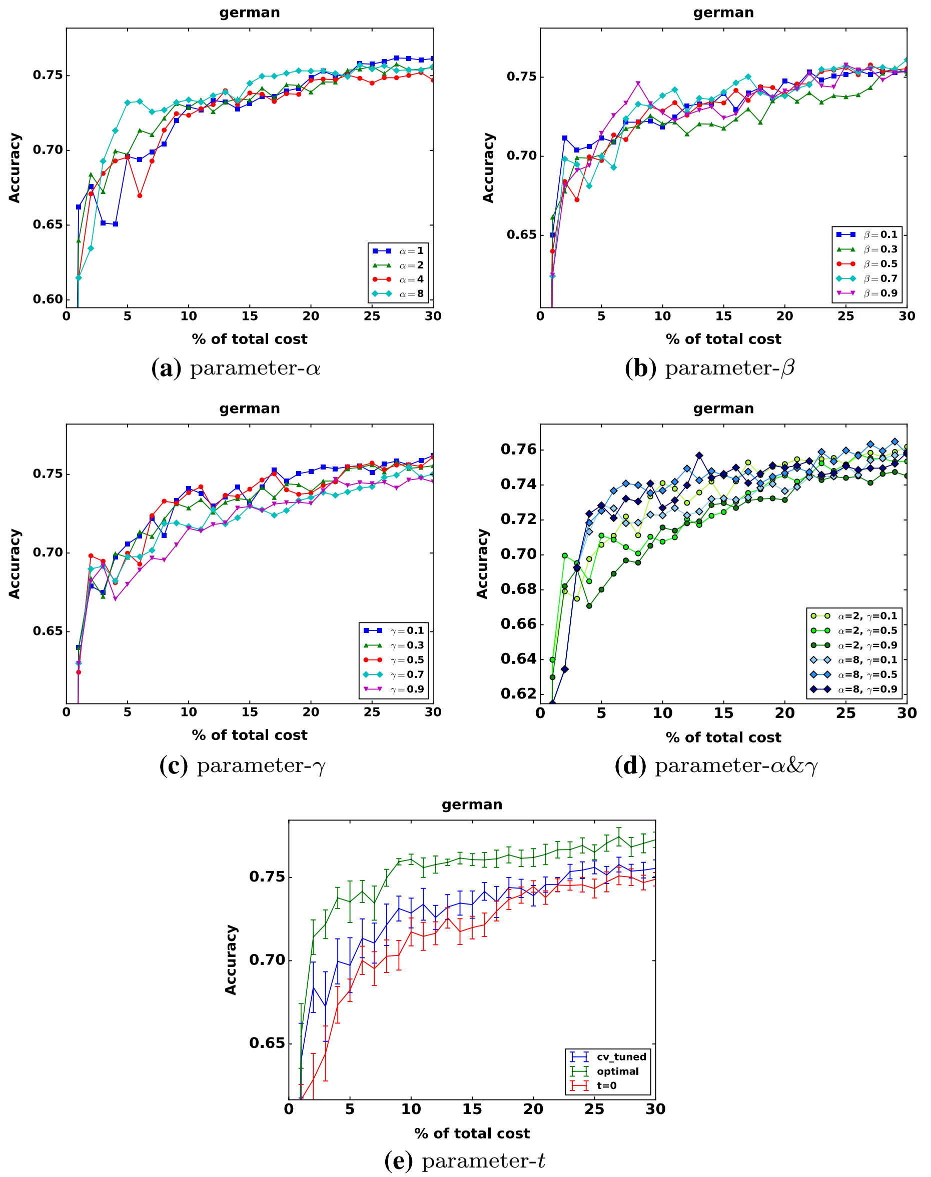
To better demonstrate the results, we only test the performance of `CSTS` on *german* dataset with the artificial *reverse distance costs*. Except for the changed parameter, others are always set as $(\alpha, \beta, \gamma) = (2, 0.5, 0.3)$ and $t \in \{0, 0.1, 0.2, 0.3\}$ is tuned by 4-fold cross validation. In parameter t, we simply compare three different settings in order to demonstrate the effectiveness of label assigning trick:

- `cv_tuned`: tuned the threshold $t \in \{0, 0.1, 0.2, 0.3\}$ by 4-fold cross validation.
- `optimal`: used the optimal threshold $t \in \{0, 0.1, 0.2, 0.3\}$ that leads to the best accuracy on testing set.
- `no_label_assigned`: set t as 0, that is, no additional data instance is labelled and added to the training dataset except the queried data instances.

4.5.2 Discussion

Figure 5 shows the experiments results on four parameters $(\alpha, \beta, \gamma, t)$. As can be seen, the performance of `CSTS` does not significantly change when different values set to parameter α, β and γ, which prove that our proposed `CSTS` algorithm is less sensitive to the parameter values and capable of providing stable performance. Some observations on the parameters' behavior:

- **parameter** α. A lower value on α can let a leaf be easier to be split, creating more clusters and leading to a higher label purity, but with a higher risk on overfitting. Therefore, when parameter α is set to a larger value 8, it could prevent `CSTS` from overfitting and reach the best performance in the early and middle stage; while $\alpha = 1$ could provide the best performance in the final stage where the number of label is large enough to lower the chance of overfitting.
- **parameter** β. A higher β makes the tree structure focusing more on the cost variance, giving a better control on the costs but losing the label purity, while a lower β has the opposite behavior. Since there is no superiority in either settings, the experiment result shows no clear trend in model performance as β increases.
- **parameter** γ. With higher γ (0.7 and 0.9), the performance of `CSTS` slightly drop owing to the low tree structure update rate.
- **parameter** α and γ. We extract two of the most sensitive parameters, α and γ, for joint analysis. The experiment result shows that when both α and γ are set to either high values (α=8, γ=0.5/0.9) or low values (α=2, γ=0.1), the algorithm has the better performance. The results indicating that when the leaves are easier to be split, the algorithm requires a higher tree structure update rate, while a low structure update rate is needed when the model is more conservative in splitting leaves.



(a) parameter-α

(b) parameter-β

(c) parameter-γ

(d) parameter-α&γ

(e) parameter-t

Fig. 5 Parameter-performance analysis

- **parameter *t*.** As can be seen, the label assigning trick performs a strong improvement on the accuracy. Notice that there is a great difference between using the optimal threshold and the cross-validation tuned threshold, indicating that the performance of CSTS can still be further improved if a better threshold parameter t is set.

In summary, changes on the parameter α, β and γ only lead to minor difference on the performance of CSTS, demonstrating that it is a stable approach in terms of the parameter sensitivity. On the other hand, the label assigning trick does enhance the model performance, while the cross-validation tuned threshold t still has a great room for improvement.

5 Conclusion

In this paper, we proposed the CSTS approach for annotation cost-sensitive active learning. The main contributions can be split to two areas: First, our proposed method is a innovative algorithm that jointly model both label and cost distribution in the supervised-built clustering structure when solving annotation cost-sensitive active learning. Furthermore, it is the first algorithm that tackles the annotation cost-sensitive active learning problems by representative sampling to the best of our knowledge. Second, the supervised tree constructing algorithm solves the issue on the impractical pure label cluster assumption in traditional representative sampling based active learning algorithms, while the overfitting problem can be delicately handled by the proposed evaluation metric. Empirical studies demonstrate the comparability and superiority of CSTS in comparing with previous methods on datasets with simulated and real annotation costs. The experiments results confirm the validity of our proposed method, and indicate CSTS is a promising approach for annotation cost-sensitive active learning. Future research should certainly further studies on label estimation considering the non-i.i.d. data distribution within a node, which could possibly be handle by other probability distributions and improve the accuracy of the estimation on the confidence interval and the label fractions. More real datasets are also welcome to better understands the behavior of CSTS in solving real world problems.

References

Arora, S., Nyberg, E., & Rosé, C. P. (2009). Estimating annotation cost for active learning in a multi-annotator environment. In *Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing, association for computational linguistics* (pp. 18–26)

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. Boca Raton: CRC Press.

Chapelle, O., Weston, J., & Schölkopf, B. (2003). Cluster kernels for semi-supervised learning. *Advances in neural information processing systems* (pp. 601–608)

Cohn, D., Atlas, L., & Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, *15*(2), 201–221.

Cuong, N., & Xu, H. (2016) Adaptive maximization of pointwise submodular functions with budget constraint. In *Advances in neural information processing systems* (pp. 1244–1252)

Dasgupta, S. (2011). Two faces of active learning. *Theoretical Computer Science*, *412*(19), 1767–1781.

Dasgupta, S., & Hsu, D. (2008). Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning* (pp. 208–215). ACM

Donmez, P., Carbonell, J. G. (2008). Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In: *Proceedings of the 17th ACM conference on Information and knowledge management* (pp 619–628). ACM

Golovin, D., & Krause, A. (2011). Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, *42*, 427–486.

Greiner, R., Grove, A. J., & Roth, D. (2002). Learning cost-sensitive active classifiers. *Artificial Intelligence*, *139*(2), 137–174.

Guillory, A., & Bilmes, J. (2009) Average-case active learning with costs. In *International conference on algorithmic learning theory* (pp 141–155). Springer

Haertel, R., Seppi, K. D., Ringger, E. K., & Carroll, J. L. (2008). Return on investment for active learning. In Proceedings of the NIPS workshop on cost-sensitive learning

Holub, A., Perona, P., & Burl, M. C. (2008) Entropy-based active learning for object recognition. In *IEEE computer society conference on computer vision and pattern recognition workshops* (pp 1–8). IEEE

Huang, K. H., & Lin, H. T. (2016) A novel uncertainty sampling algorithm for cost-sensitive multiclass active learning. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*

Huang, S. J., Jin, R., & Zhou, Z. H. (2010) Active learning by querying informative and representative examples. In *Advances in neural information processing systems* (pp. 892–900)

Huang, S. J., Chen, J. L., Mu, X., & Zhou, Z. H. (2017) Cost-effective active learning from diverse labelers. In *Proceedings of the 26th international joint conference on artificial intelligence* (pp 1879–1885). AAAI Press

Kang, J., Ryu, K. R., & Kwon, H. C. (2004) Using cluster-based sampling to select initial training set for active learning in text classification. In *Pacific-Asia conference on knowledge discovery and data mining* (pp 384–388). Springer.

King, R. D., Whelan, K. E., Jones, F. M., Reiser, P. G., Bryant, C. H., Muggleton, S. H., et al. (2004). Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, *427*(6971), 247–252.

Lewis, D. D., & Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval* (pp 3–12). New York: Springer.

Lichman, M. (2013). UCI machine learning repository. http://archive.ics.uci.edu/ml

Liu, A., Jun, G., & Ghosh, J. (2009) Spatially cost-sensitive active learning. In *Proceedings of the 2009 SIAM international conference on data mining* (pp 814–825). SIAM

Liu, Y. (2004). Active learning with support vector machine applied to gene expression data for cancer classification. *Journal of Chemical Information and Computer Sciences*, *44*(6), 1936–1941.

Margineantu, D. D. (2005). Active cost-sensitive learning. In *Proceedings of international joint conference on artificial intelligence* (pp 1622–1623)

Nguyen, H.T., & Smeulders, A. (2004) Active learning using pre-clustering. In *Proceedings of the 21th international conference on Machine learning* (p. 79). ACM.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106.

Quinlan, J. R. (2014). *C4.5: Programs for machine learning*. Amsterdam: Elsevier.

Ringger, E., McClanahan, P., Haertel, R., Busby, G., Carmen, M., Carroll, J., Seppi, K., Lonsdale, D. (2007). Active learning for part-of-speech tagging: Accelerating corpus annotation. In *Proceedings of the Linguistic Annotation Workshop, Association for Computational Linguistics* (pp 101–108)

Seeger, M. (2000). *Learning with labeled and unlabeled data*. Tech. rep., technical report, University of Edinburgh

Settles, B. (2010). *Active learning literature survey* (p. 11). Madison: University of Wisconsin.

Settles, B., Craven, M., & Friedland, L. (2008) Active learning with real annotation costs. In *Proceedings of the NIPS workshop on cost-sensitive learning* (pp 1–10)

Tomanek, K., & Hahn, U. (2010). A comparison of models for cost-sensitive active learning. In *Proceedings of the 23rd international conference on computational linguistics: Posters, association for computational linguistics* (pp 1247–1255)

Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, *2*(Nov), 45–66.

Vapnik, V. (2013). *The nature of statistical learning theory*. Berlin: Springer.

Vijayanarasimhan, S., & Grauman, K. (2011). Cost-sensitive active visual category learning. *International Journal of Computer Vision*, *91*(1), 24–44.

Xu, Z., Yu, K., Tresp, V., Xu, X., & Wang, J. (2003) Representative sampling for text classification using support vector machines. In *European conference on information retrieval* (pp. 393–407). Springer.

Yan, Y., & Huang, S. J. (2018) Cost-effective active learning for hierarchical multi-label classification. In *IJCAI* (pp. 2962–2968)