One-Class Transfer Learning with Uncertain Data

Bo Liu[1,⋆], Philip S. Yu[2,3], Yanshan Xiao[4], and Zhifeng Hao[4]

[1] Department of Automation, Guangdong University of Technology
csbliu@gmail.com
[2] Department of Computer Science, University of Illinois at Chicago
[3] Computer Science Department, King Abdulaziz University
psyu@uic.edu
[4] Department of Computer Science, Guangdong University of Technology
xiaoyanshan@gmail.com, mazfhao@scut.edu.cn

Abstract. One-class learning aims at constructing a distinctive classifier based on the labeled one class data. However, it is a challenge for the existing one-class learning methods to transfer knowledge from a source task to a target task for uncertain data. To address this challenge, this paper proposes a novel approach, called uncertain one-class transfer learning with SVM (UOCT-SVM), which first formulates the uncertain data and transfer learning into one-class SVM as an optimization problem and then proposes an iterative framework to build an accurate classifier for the target task. Our proposed method explicitly addresses the problem of one-class transfer learning with uncertain data. Extensive experiments has found our proposed method can mitigate the effect of uncertain data on the decision boundary and transfer knowledge to help build an accurate classifier for the target task, compared with state-of-the-art one-class learning methods.

Keywords: Transfer Learning, Data of Uncertainty, One-class Learning.

1 Introduction

One-class learning has been proposed to handle the case where only one class of data is labeled in the training phase [19,17]. In this case, the labeled class of data is called *target class*, while all other samples not in this class are called the *non-target class*. In some real-world applications, such as anomaly detection [5,21], it is easy to obtain one class of normal data, whereas collecting and labeling abnormal instances may be expensive or impossible. To date, one-class learning has been found in a large variety of applications from anomaly detection [5], automatic image annotation [11], to sensor data drift detection [18].

The previous one-class learning can be classified into two broad categories: (1) the methods for one-class learning with unlabeled data [13,12,25,4,24], in which they first extracts negative examples from the unlabeled data, and then constructs a binary classifier based on the labeled target class and the extracted

⋆ Corresponding author.

J. Pei et al. (Eds.): PAKDD 2013, Part I, LNAI 7818, pp. 471–483, 2013.

negative class. For example, the method in [25] first uses a 1-DNF technique to extract negative documents and utilize SVM to iteratively build a binary classifier. (2) the method for one-class learning without unlabeled data [17,14], in which one-class SVM first maps the target data into a feature space and then constructs a hyper-plane to separate the target class and the origin of the feature space. The learned classifier is then utilized to classify a test sample into target class or non-target class.

Despite much progress on the one-class learning, most of the previous work considers the one-class learning as a single learning task. However, in many real-world applications, we expect to reduce the labeling effort of a new task (referred to as target task) by transferring knowledge from the related task (source task), which is called transfer learning [15]. For example, we may have plenty of user's previously labeled documents, which indicate the users' interest; as time goes on, user's interest may gradually drift; however, we may not have too much user's currently labeled documents, since labeling plenty of documents timely may be impossible for the user. Therefore, we expect the user's previously labeled documents can transfer knowledge to help build an one-class classifier for the target task. Another important observation is that, collected data in many real-world applications is uncertain in nature [2]. This is because data collection methodologies are only able to capture a certain level of information, making the extracted data incomplete or inaccurate [2]. For example, in environmental monitoring applications, sensor networks typically generate a large amount of uncertain data because of instrument errors, limited accuracy or noise-prone wireless transmission [2]. Therefore, it is necessary to develop the one-class transfer learning method for uncertain data, and build an accurate classifier by transferring knowledge from the source task to the target task for prediction.

This paper addresses the problem of one-class transfer learning with uncertain data. To build an one-class transfer learning classifier for uncertain data, we have two challenges. The first one is to formulate data of uncertainty and transfer learning into the one-class learning. The second is to solve the formulated optimization to build an one-class classifier for the target task. To handle the above challenge, we propose a novel approach, called uncertain one-class transfer learning with SVM (UOCT-SVM), which incorporates data uncertainty and knowledge transfer into one-class SVM and provides an efficient framework to build an one-class classifier for the target task. The contribution of our work can be summarized as follows.

1. We incorporate the transfer learning and uncertain data into the one-class SVM such that the transferred knowledge can benefit the one-class classifier for the target task. To handle uncertain data, we introduce the bound score into the learning to relocate the uncertain data and refine the decision boundary.
2. We propose the usage of an iterative framework to mitigate the effect of noise on the one-class classifier and transfer knowledge from the source task to the target task. To the best of our knowledge, this is the first work to

explicitly handle data uncertainty and knowledge transfer in the one-class learning.

3. We conduct extensive experiments to evaluate the performance of our UOCT-SVM method. The results show that our UOCT-SVM can mitigate the effect of noise on the decision boundary and transfer knowledge to help build an accurate classifier for the target task compared with state-of-the-art one-class learning methods.

Section 2 discusses the related work. Section 3 introduces the preliminaries. Section 4 presents our proposed approach. Section 5 reports experimental results. Section 6 concludes the paper and future work.

2 Related Work

In this section, we briefly review previous work related to our study.

2.1 Mining Uncertain Data

In data collection, some records in the data might be degraded due to noise, precision of equipment, and are considered uncertain in their representation [2]. We briefly review the previous work on uncertain data as follows.

For the clustering and classification methods with uncertain data, they develop on the clustering and classification methods. FOPTICS [9] introduces a fuzzy distance function to measure the similarity between uncertain data on top of the hierarchical density-based clustering algorithm. The method in [8] studies the problem of clustering uncertain objects whose locations are described by probability density functions to cluster uncertain data. In addition, binary SVM is extended to handle uncertain data [7] to provide a geometric algorithm.

2.2 Transfer Learning

In transfer learning [15], the knowledge is expected to transfer from a source task into the learning of target task such that the transferred knowledge can benefit the learned classifier for the target task. We briefly review some of them as follows.

The work in [10] assumes the distribution of target and source tasks fit the Gaussian process. However, it assumes the distribution of the data to be specified as a priori, which makes them inapplicable to many real-world applications. Other algorithms such as [16] assume that some instances or features can be used as a bridge for knowledge transfer.

Multi-task learning [20] is closely related to transfer learning. In multi-task learning, several tasks are learned simultaneously. In contrast to multi-task learning, transfer learning focuses on transferring knowledge from the source task to the target task, rather than ensuring the performance of each task.

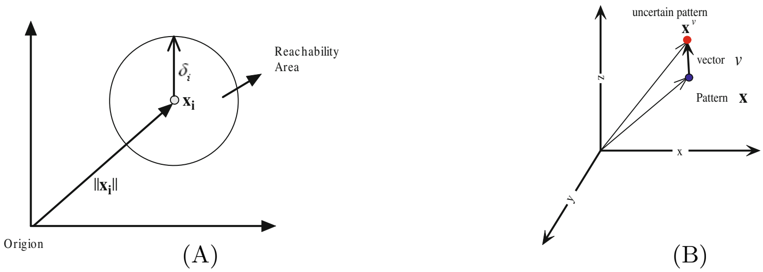

Fig. 1. (A): Illustration of reachability area of instance $\mathbf{x}_i$. (B): Illustration of the method used to add the noise to a data example: $\mathbf{x}$ is an original data example, $\mathbf{v}$ is a noise vector, $\mathbf{x}^\mathbf{v}$ is the new data example with added noise. Here we have $\mathbf{x}^\mathbf{v} = \mathbf{x} + \mathbf{v}$. .

3 Preliminary

3.1 One-Class SVM

Suppose the training target class is $S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{|S|}\}$, where $\mathbf{x}_i \in \ R^n$. In one-class SVM, input data is mapped from the input space into a feature space and the inner product of two vectors $\phi(\mathbf{x})$ and $\phi(\mathbf{x}_i)$ can be calculated by a kernel function $K(\mathbf{x}, \mathbf{x}_i) = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}_i)$. One-class SVM aims to determine a hyperplane to separate the target class and the origin of the space:

$$\begin{aligned} &min\ \tfrac{1}{2} \parallel \mathbf{w} \parallel^2 - \rho + C \textstyle\sum_{i=1}^{|S|} \xi_i \\ &s.t. \qquad \mathbf{w} \cdot \mathbf{x}_i \geq \rho - \xi_i \\ &\qquad \xi_i \geq 0, \quad i = 1, 2, \ldots, |S|, \end{aligned} \tag{1}$$

where $\mathbf{w}$ is vector, parameter C is used to tradeoff the sphere volume and the errors. After solve problem (1) and obtain $\mathbf{w}$ and $\rho = \mathbf{w} \cdot \phi(\mathbf{x})$. For a test sample $\mathbf{x}_t$, if $\mathbf{w} \cdot \phi(\mathbf{x}_t) > \rho$, it is classified into the target class; otherwise, it belongs to the non-target class.

In this paper, we extend the standard one-class SVM for one-class transfer learning with data of uncertainty.

3.2 Uncertain Model

For the labeled target class, we assume each input data $\mathbf{x}_i$ is subject to an additive noise vector $\triangle\mathbf{x}_i$. In this case, the original uncorrupted input $\mathbf{x}_i^s$ is denoted $\mathbf{x}_i^s = \mathbf{x}_i + \triangle\mathbf{x}_i$. We can assume $\triangle\mathbf{x}_i$ follows a given distribution. The method of bounded and ellipsoidal uncertainties has been investigated in [6,14]. In this situation, we consider a simple bound score for each instance such that $\parallel \triangle\mathbf{x}_i \parallel \leq \delta_i$.

We then let $\mathbf{x}_i + \triangle\mathbf{x}_i$ ($\parallel \triangle\mathbf{x}_i \parallel \leq \delta_i$) denote the *reachability area* of instance $\mathbf{x}_i$ as illustrated in Figure 1. (A). We then have

$$\parallel \mathbf{x}_i^s \parallel = \parallel \mathbf{x}_i + \triangle\mathbf{x}_i \parallel \leq \parallel \mathbf{x}_i \parallel + \parallel \triangle\mathbf{x}_i \parallel \leq \parallel \mathbf{x}_i \parallel + \delta_i \tag{2}$$

In this way, $\mathbf{x}_i^s$ falls in the reachability area of $\mathbf{x}_i$. By using the bound score for each input sample, we can convert the uncertain one-class transfer learning into standard one-class learning with constraints.

4 One-Class Transfer Learning on Uncertain Data

In this section, we put forward our one-class transfer learning to handle uncertain data. Suppose we have two tasks, that is, to train one-class classifier on S_s for source task and on S_t for target task. Let

$$w_1 = w_o + v_1 \quad and \quad w_2 = w_o + v_2, \tag{3}$$

where w_1 and w_2 are parameters of the one-class SVM for source and target tasks, respectively. w_o is a common parameter while v_1 and v_2 are specific parameters. Here, w_o can be considered as a bridge to transfer knowledge from source task to the target task [20]. By assuming $\rho_1 = w_1 \cdot \mathbf{x}$ and $\rho_2 = w_2 \cdot \mathbf{x}$ to be two hyperplanes for S_s and S_t respectively, w_1 and w_2 can be denoted as $w_t = w_0 + v_t, t = 1, 2$ and the extended version of one-class transfer learning for uncertain data can be written as follows.

$$\begin{aligned} min \quad & \tfrac{1}{2}\|w_o\|^2 + \textstyle\sum_{t=1}^{2} C_t\|v_t\|^2 - \rho_1 - \rho_2 + C(\sum_{\mathbf{x}_i \in S_s} \xi_i + \sum_{\mathbf{x}_j \in S_t} \xi_j) \\ s.t. \quad & (w_o + v_1) \cdot (\mathbf{x}_i + \triangle\mathbf{x}_i) \geq \rho_1 - \xi_i, \qquad \mathbf{x}_i \in S_s \\ & (w_o + v_2) \cdot (\mathbf{x}_j + \triangle\mathbf{x}_j) \geq \rho_2 - \xi_j, \quad \xi_j \geq 0 \quad \mathbf{x}_j \in S_t \\ & \xi_i \geq 0, \quad \xi_j \geq 0, \quad \|\triangle\mathbf{x}_i\| \leq \delta_i, \quad \|\triangle\mathbf{x}_j\| \leq \delta_j. \end{aligned} \tag{4}$$

For the above optimization, we then have:

1 $\mathbf{x}_i + \triangle\mathbf{x}_i$ and $\mathbf{x}_j + \triangle\mathbf{x}_j$ denote the original vectors which are affected by $\triangle\mathbf{x}_i$ and $\triangle\mathbf{x}_j$. Thus, one-class transfer learning classifier can be less sensitive to the sample corrupted by noise since we can always determine a choice of $\triangle\mathbf{x}_i$ to render $\mathbf{x}_i + \triangle\mathbf{x}_i$ to refine the one-class transfer decision boundary. $\|\triangle\mathbf{x}_i\| \leq \delta_i$ and $\|\triangle\mathbf{x}_j\| \leq \delta_j$ restrict the range of the uncertain information by a bound score, which has been utilized in previous work [14].

2 We utilize common parameter w_o as a bridge to transfer knowledge from source task to target task. Parameters C_1 and C_2 control the preference of the two tasks. If $C_1 > C_2$, task 1 is preferred to task 2; otherwise, task 2 is preferred to task 1. Parameters ξ_i and ξ_j are defined as measures of error.

4.1 Solution to Uncertain One-Class Transfer Learning Classifier

As the above optimization problem (4) is far more complicated than the standard one-class SVM, we will use an iterative approach to calculate ρ_1, ρ_2, $\triangle\mathbf{x}_i$ and $\triangle\mathbf{x}_j$ such that we can obtain the one-class transfer learning classifier for uncertain data. The iterative steps can be summarized as follows. (a): fix each $\triangle\mathbf{x}_i$ and $\triangle\mathbf{x}_j$ to solve the problem (4) to obtain $\overline{\rho}_1$ and $\overline{\rho}_2$; (b): fix the obtained $\overline{\rho}_1$ and $\overline{\rho}_2$ to calculate $\triangle\overline{\mathbf{x}}_i$ and $\triangle\overline{\mathbf{x}}_j$ iteratively. We detail the alternating two steps as follows. (We omit the detailed derivation of the Theorems in this section due to space limitation)

Calculation of Classifier by Fixing $\triangle\ \overline{\mathbf{x}}_i$ and $\triangle\ \overline{\mathbf{x}}_j$. First of all, we fix each $\triangle\ \overline{\mathbf{x}}_i$ and $\triangle\ \overline{\mathbf{x}}_j$ as small values such that $\| \triangle\overline{\mathbf{x}}_i \| < \delta_i$, $\| \triangle\overline{\mathbf{x}}_j \| < \delta_j$ [1]. Based on this, the constrains $\| \triangle\overline{\mathbf{x}}_i \| < \delta_i$, $\| \triangle\overline{\mathbf{x}}_j \| < \delta_j$ in problem (4) won't have effect on the solution. Then, problem (4) is equivalent to

$$\begin{aligned} min \quad & \tfrac{1}{2}\|w_o\|^2 + \textstyle\sum_{t=1}^{2} C_t\|\upsilon_t\|^2 - \rho_1 - \rho_2 + C(\sum_{\mathbf{x}_i \in S_s} \xi_i + \sum_{\mathbf{x}_j \in S_t} \xi_j) \\ & s.t. \ \ (w_o + \upsilon_1) \cdot (\mathbf{x}_i + \triangle\overline{\mathbf{x}}_i) \geq \rho_1 - \xi_i, \quad \xi_i \geq 0 \quad \mathbf{x}_i \in S_s \\ & \qquad (w_o + \upsilon_2) \cdot (\mathbf{x}_j + \triangle\overline{\mathbf{x}}_j) \geq \rho_2 - \xi_j, \quad \xi_j \geq 0 \quad \mathbf{x}_j \in S_t \end{aligned} \tag{5}$$

We then have the following Theorem.

Theorem 1: By using Lagrangian function [22], the solution of the optimization problem (5) is to solve the following dual problem

$$\begin{aligned} F(\alpha) = & \tfrac{1}{2}\|w_o\|^2 + C_1\|\upsilon_1\|^2 + C_2\|\upsilon_2\|^2 - \textstyle\sum_{\mathbf{x}_i \in S_s} \alpha_i[(w_0 + \upsilon_1) \cdot \overline{\mathbf{x}}_i] \\ & - \textstyle\sum_{\mathbf{x}_j \in S_t} \alpha_j[(w_0 + \upsilon_2) \cdot \overline{\mathbf{x}}_j] \\ s.t. \quad & 0 \leq \alpha_i \leq C, \ \ 0 \leq \alpha_j \leq C, \quad \textstyle\sum_{\mathbf{x}_i \in S_s} \alpha_i = 1, \ \ \sum_{\mathbf{x}_j \in S_t} \alpha_j = 1, \end{aligned} \tag{6}$$

in which

$$w_o = \textstyle\sum_{\mathbf{x}_i \in S_s} \alpha_i \cdot \overline{\mathbf{x}}_i + \sum_{\mathbf{x}_j \in S_t} \alpha_j \cdot \overline{\mathbf{x}}_j,$$
$$\upsilon_1 = \tfrac{1}{2} \textstyle\sum_{\mathbf{x}_i \in S_s} \alpha_i \cdot \overline{\mathbf{x}}_i, \quad \upsilon_2 = \tfrac{1}{2} \sum_{\mathbf{x}_j \in S_t} \alpha_j \cdot \overline{\mathbf{x}}_j,$$

where α_i and α_j are the Lagrange multipliers and $\overline{\mathbf{x}}_i = \mathbf{x}_i + \triangle\mathbf{x}_i$, $\overline{\mathbf{x}}_j = \mathbf{x}_j + \triangle\mathbf{x}_j$. After solve optimization problem (6), we obtain α_i and α_j, and $\overline{\rho}_1 = (w_0 + \upsilon_1) \cdot \mathbf{x}_i$ and $\overline{\rho}_2 = (w_0 + \upsilon_2) \cdot \mathbf{x}_j$.

Calculation of $\triangle\ \overline{\mathbf{x}}_i$ and $\triangle\ \overline{\mathbf{x}}_j$ by Fixing the Classifier. After setting $\triangle\overline{\mathbf{x}}_i$ and $\triangle\overline{\mathbf{x}}_j$ as a small values which are less than δ_i and δ_j respectively, and solving optimization problem (6), we obtain $\overline{\rho}_1$ and $\overline{\rho}_2$. The next step is to use the obtained $\overline{\rho}_1$ and $\overline{\rho}_2$ to calculate new $\triangle\overline{\mathbf{x}}_i$ and $\triangle\overline{\mathbf{x}}_j$. We then have Theorem 2 as follows.

Theorem 2: If the hyperplanes for the source and target tasks are denoted as $\overline{\rho}_1 = (\overline{w}_0 + \overline{\upsilon}_1) \cdot \mathbf{x}$, and $\overline{\rho}_2 = (\overline{w}_0 + \overline{\upsilon}_2) \cdot \mathbf{x}$, the solution of problem (4) over $\triangle\overline{\mathbf{x}}_i$ and $\overline{\mathbf{x}}_j$ are

$$\triangle\overline{\mathbf{x}}_i = \delta_i \frac{\overline{w}_0 + \overline{\upsilon}_1}{\| \overline{w}_0 + \overline{\upsilon}_1 \|}, \tag{7}$$

$$\triangle\overline{\mathbf{x}}_j = \delta_j \frac{\overline{w}_0 + \overline{\upsilon}_2}{\| \overline{w}_0 + \overline{\upsilon}_2 \|}. \tag{8}$$

This Theorem indicates that, for a given $\overline{\rho}_1$ and $\overline{\rho}_2$, the minimization of problem (4) over $\triangle\overline{\mathbf{x}}_i$ and $\triangle\overline{\mathbf{x}}_j$ is quite straightforward.

After that, we have one round of alternation and continue to update $\overline{\rho}_1$, $\overline{\rho}_2$, $\triangle\overline{\mathbf{x}}_i$, $\triangle\overline{\mathbf{x}}_2$ until the algorithm converges.

[1] We set $\triangle\ \overline{\mathbf{x}}_i = 0$ and $\triangle\ \overline{\mathbf{x}}_j = 0$ in the first step of the iterative framework.

Algorithm 1. Uncertain one-class transfer learning with uncertain data

```
Input: S_s, S_t ; // source and target tasks
.......... C_1, C_2 and C. // parameters
.......... δ_i, δ_j // bound value for samples in both tasks.
Output: ρ̄_1 and ρ̄_2.
Initialize each △x̄_i = 0 and △x̄_j = 0;
t=0;
Initialize F_va(t) = ∞;
repeat
    t = t + 1;
    Fix △x̄_i for i = 1, 2, ..., |S_s| and △x̄_j for j = 1, 2, ..., |S_t| to solve problem (5);

    Let F_va(t) = F(α);
    Obtain α_i, i = 1, 2, ..., |S_s|;
    Obtain α_j, j = 1, 2, ..., |S_t|;
    Obtain the hyperplane ρ̄_1 = (w̄_0 + v̄_1) · x for source task;
    Obtain the hyperplane ρ̄_2 = (w̄_0 + v̄_2) · x for the target task;
    Fix ρ̄_1 and ρ̄_1 to update each △x̄_i and △x̄_j according to Equation (7) and (8);
until |F_va(t) − F_va(t − 1)| < ε|F_va(t − 1)|
Return ρ̄_1 = (w̄_0 + v̄_1) · x and ρ̄_2 = (w̄_0 + v̄_2) · x.
```

To utilize Theorem 1 and Theorem 2 iteratively to calculate $\overline{\rho}$ and $\triangle\overline{\mathbf{x}}$, we have Theorem 3 as follows.

Theorem 3: If optimal $\triangle\overline{\mathbf{x}}_i = \delta_i \frac{\overline{w}_0+\overline{v}_1}{\|\overline{w}_0+\overline{v}_1\|}$ and $\triangle\overline{\mathbf{x}}_j = \delta_j \frac{\overline{w}_0+\overline{v}_2}{\|\overline{w}_0+\overline{v}_2\|}$ are fixed, the solution of problem (4) is equivalent to optimization problem (6).

From $\triangle\overline{\mathbf{x}}_i = \delta_i \frac{\overline{w}_0+\overline{v}_1}{\|\overline{w}_0+\overline{v}_1\|}$, we have $\|\triangle\overline{\mathbf{x}}_i\| = \delta_i \frac{\|\overline{w}_0+\overline{v}_1\|}{\|\overline{w}_0+\overline{v}_1\|} = \delta_i$, then the constrains $\triangle\overline{\mathbf{x}}_i \leq \delta_i$ in problem (4) won't have any effect on problem (4). The same analysis can be used to $\triangle\overline{\mathbf{x}}_j = \delta_j \frac{\overline{w}_0+\overline{v}_2}{\|\overline{w}_0+\overline{v}_2\|}$. Thus, problem (4) equals to problem (6).

Iterative Framework. So far, we have introduced the framework to update $\overline{\rho}_1$, $\overline{\rho}_2$, $\triangle\overline{\mathbf{x}}_i$ and $\triangle\overline{\mathbf{x}}_j$ at a round, and we can use the above steps to obtain an uncertain one-class transfer learning classifier. By referring to the alternating optimization method in [6], we propose the usage of the iterative approach to solve problem (4) in Algorithm 1.

In Algorithm 1, ε is a threshold. Since the value of $F_{val}(t)$ is nonnegative, with the decreasing of $F_{val}(t)$, $|F_{val}(t) - F_{val}(t-1)|/|F_{val}(t-1)|$ will be smaller than a threshold. Thus, Algorithm 1 can converge in finite steps.

After that, we obtain the uncertain one-class transfer learning classifiers for the target task. We then utilize the learned classifier for prediction.

Note:(1): For the determination of δ_i for the sample $\mathbf{x}_i$ in S_s, we calculate the average distance of $\mathbf{x}_i$ between it and the its $k-$nearest neighbors. The same operation is utilized to the sample $\mathbf{x}_j$ in S_t. This setting is previously utilized in the previous work [14]. At the beginning of the framework, we initialize each $\triangle\overline{\mathbf{x}}_i = 0$, $\triangle\overline{\mathbf{x}}_j = 0$ and update them base on (7) and (8). Then, we can have

$\triangle\overline{\mathbf{x}}_i \leq \delta_i$ and $\triangle\overline{\mathbf{x}}_j \leq \delta_j$. (2): Above, we present the formulation of uncertain one-class transfer learning in the input space; while for the kernel space, we can utilize $K(\mathbf{x}, \mathbf{q}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{q})$ in the above formulation.

5 Experiments

5.1 Baseline and Metrics

In this section, we investigate the performance of our proposed UOCT-SVM method. In transfer learning, we expect the transferred knowledge from the source task to the target task can improve the performance of the classifier built on the target task. For comparison, another two methods are used as baselines.

1. The first method is the standard one-class SVM (OC-SVM), which determines a hyperplane to separate the target class and the origin of feature space. This baseline is used to show the improvement of our method over the standard one-class SVM.
2. The second baseline is the uncertain one-class SVM (UOC-SVM) [14], which builds one-class classifier on uncertain data. This baseline is utilized to investigate the ability of transferred knowledge contributed to the construction of classifier on the target data.

The performance of classification systems is typically evaluated in terms of F-measure [23], we use it as metrics. The F measure trades off precision p and recall r: $F = 2p \cdot r/(p + r)$. From the definition, we know only when both precision and recall are large, will the F-measure exhibit large value.

5.2 Dataset and Experiment Setting

One-Class Learning Data. To evaluate the properties of our approach, we conduct experiments on 20 Newsgroups [2] and Reuters-21578 [3]. Both data sets have hierarchical structures. The 20 Newsgroups corpus contains several top categories, and under the top categories, there are 20 sub-categories where each subcategory has 1000 samples. Similarly, Reuters-21578 contains Reuters news wire articles organized into five top categories, and each category includes different sub-categories.

Following the previous work [17,19] for one-class learning, we reorganize the original data in a way for the one-class transfer learning problem as follows. For the 20 Newsgroups, we consider one sub-category as target class, and select a number of example from other categories as non-target class. Specifically, we first choose a sub-category (a_1) from a top category (A), and consider this sub-category (a_1) as the target class and consider the examples from other top categories, i.e., except for category (A) as non-target class. Based on the this, we generate target class and non-target class for the source task. For the target

[2] Available at `http://people.csail.mit.edu/jrennie/20Newsgroups/`
[3] Available at `http://www.daviddlewis.com/resources/testcollections/`

task, we choose a sub-category (a_2) from the same top category (A) as that for the source task, and consider this sub-category (a_2) as the target class; while take the examples from other top categories except for (A) as non-target class.

For the Reuters-21578, each top category has many sub-categories, for example, "people" has 267 sub-categories and the size of each sub-category is not always large. We organize it as follows. For a top category (A), all of the subcategories are organized into two parts (denoted as $a(1)$ and $a(2)$), and each part is approximately equal in size. We then regard $a(1)$ and $a(2)$) as the target class of the source task and target task respectively. We also consider the examples from the other category except for (A) as the non-target class for the source and target task respectively.

In the above operations, we generate target class from the same top category (A), that are $a(1)$ and $a(2)$, for the source task and target task, this is because we should guarantee the two tasks are related. Otherwise, the transfer learning may not, and may even hurt, the performance of a target task, which can be referred to as negative transfer [3].

Uncertain Information Generation. We note that the above data are deterministic, so we need to model and involve uncertainty to these data sets. Following the method in the previous work [1], we generate the uncertain data as follows.

For generate data, we first compute the standard deviation σ_i^0 of the entire data along the ith dimension, and then obtain the standard deviation of the Gaussian noise σ_i randomly from the range $[0, 2\cdot\eta\cdot\sigma_i^0]$. For the ith dimension, we add noise from a random distribution with standard deviation σ_i. Thus, a data example $\mathbf{x}_j$ is added with the noise, i.e., $\sigma^{\mathbf{x}_j} = [\sigma_1^{\mathbf{x}_j}, \sigma_2^{\mathbf{x}_j}, \cdots, \sigma_{r-1}^{\mathbf{x}_j}, \sigma_r^{\mathbf{x}_j}]$. Here, r denotes the number of dimensions for a data example $\mathbf{x}_j$, and $\sigma_i^{\mathbf{x}_j}$, $i = 1, \cdots r$ represents the noise added into the ith dimension of the data example. Fig. 1 (B) illustrates the basic idea of the method.

In the experiment, RBF kernel function ($K(\mathbf{x}, \mathbf{x}_i) = exp(-\| \mathbf{x} - \mathbf{x}_i \|_2^2/2\sigma^2)$) is used in the experiment since it is the most common kernel function. The σ in RBF kernel function is ranged from 2^{-10} to 2^{10}. In our method, C_1 and C_2 control the tradeoff between the source task and target task. Since we care about the target task more than the source task, we set $C_2 > C_1$ and C_1, C_2, C is chosen from 1 to 1000. For the k-nearest neighbors to generate bound score, we set k equal to ten percent of the training target class. We set ε is set as 0.15 in the experiment. All the experiments are on a laptop with a 2.8 GHz processor and 3GB DRAM.

Performance Comparison. For the target data set, we randomly choose around 20% to form a training set while the remaining example are used for testing. This is because transfer learning always assumes we do not have sufficient training data for the target task. We also conduct 10-fold cross validation on the test set. For the source data, since we are more concerned about the performance of the target task we incorporate around 80% them into training and

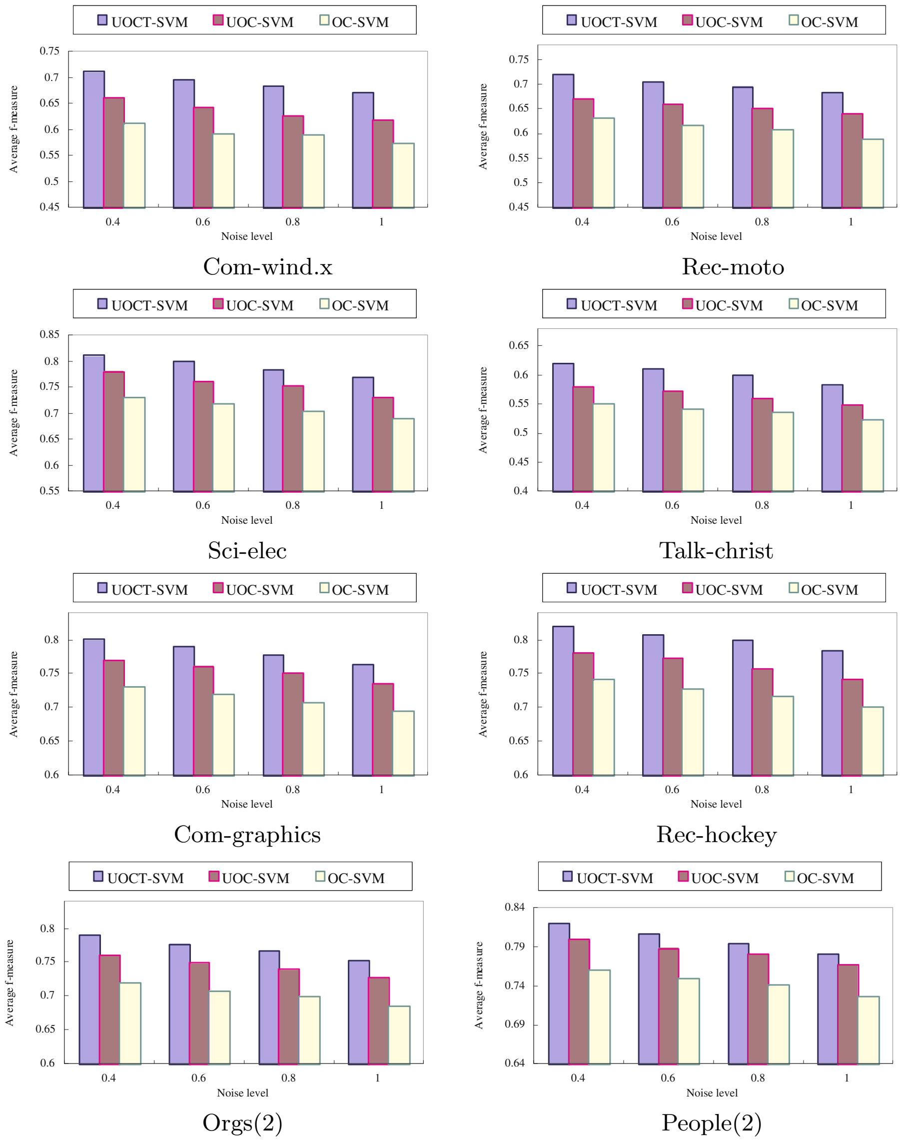

Fig. 2. The performance of OC-SVM, UOC-SVM and UOCT-SVM at different noise level

Table 1. The average F-measure accuracy and standard deviation for the target task obtained by OC-SVM, UOC-SVM and UOCT-SVM methods

Number	Source task	Target task	OC-SVM	UOC-SVM	UOCT-SVM
1	Com-misc	Com-wind.x	0.61 ± 0.042	0.66 ± *0.035*	*0.71* ± *0.035*
2	Rec-autos	Rec-moto	0.62 ± 0.034	0.67 ± 0.031	*0.72* ± *0.029*
3	Sci-med	Sci-elec	0.73 ± 0.053	0.78 ± *0.042*	*0.81* ± *0.042*
4	Talk-religion	Talk-christ	0.53 ± 0.055	0.58 ± 0.052	*0.62* ± *0.049*
5	Com-wind.misc	Com-graphics	0.71 ± 0.045	0.77 ± *0.042*	*0.80* ± *0.042*
6	Rec-baseball	Rec-hockey	0.72 ± 0.032	0.78 ± 0.031	*0.82* ± *0.030*
7	Orgs(1)	Orgs(2)	0.70 ± 0.051	0.76 ± 0.049	*0.79* ± *0.045*
8	People(1)	People(2)	0.74 ± 0.047	0.80 ± *0.035*	*0.82* ± 0.042

the remaining are used for testing. To avoid a sampling bias, we repeat the above process 10 times, and report the average f-measure accuracy and the standard deviations in Table 1, in which we set the noise level at 0.4.

It can be seen that, our proposed UOCT-SVM method always provides a superior performance compared with UOC-SVM. Although both UOCT-SVM and UOC-SVM can handle data of uncertainty, our method can transfer knowledge from the source task to the target task such that we can develop an accurate classifier for the target task. In addition, both UOCT-SVM and UOC-SVM perform much better than the standard OC-SVM, this occurs because UOCT-SVM and UOC-SVM reduce the effect of the noise on the decision boundary; as results, they can deliver better one-class classifier compared with the standard OC-SVM. In addition, we find the standard deviation of our method is less than the UOC-SVM and OC-SVM for most data sets.

Performance on Different Noise Levels. We investigate the performance sensitivity of three methods on different noise level from 0.4 to 1. In Fig. 2, we illustrate the variation in effectiveness with increasing noise error. On the $x-$axis, we illustrate the noise level. On the $y-$axis, we illustrate the average f-measure value. It is clear that in each case, the f-measure value reduces with the increasing noise level. This occurs because when the level of noise increases, the target class potentially becomes less distinguishable from the non-target class. However, we can clearly see that, UOCT-SVM approach can still consistently yield higher f-measure value than OC-SVM and UOC-SVM. This indicates that, UOCL method can reduce the effect of noise. In addition, UOC-SVM performs better than OC-SVM since UOC-SVM can reduce the effect of noise on the decision classifier.

Average Running Time Comparison. So far, we have investigated the performance of the three methods, it is interesting to compare the running time of them. The average running time of OC-SVM, UOC-SVM and UOCT-SVM are 1553, 4763 and 6980 seconds respectively. We find that the standard one-class SVM performs much faster than UOC-SVM and UOCT-SVM since

OC-SVM does not consider the data of uncertainty and transfer knowledge in the learning; as a result, it performs faster while has the lowest accuracy compared with UOC-SVM and UOCT-SVM. In addition, UOC-SVM proceeds faster than UOCT-SVM, since the latter one transfers knowledge from the source task to the target task to benefit the classifier for target task, which takes time to fulfill the knowledge transfer.

6 Conclusion and Future Work

This paper proposes a novel approach, called UOCT-SVM, for one-class transfer learning with uncertain data. Our proposed UOCT-SVM first formulates the uncertain data and transfer learning to the one-class SVM learning as an optimization problem, and then puts forward an efficient framework to solve the optimization problem such that we can obtain an accurate classifier for the target task by transferring knowledge from the source task to the target task. Extensive experiment has investigated the performance of our proposed UOCT-SVM.

In the future, we would like to investigate how to design better methods to generate bound scores based on the data characteristics in a given application domain.

Acknowledgements. This work is supported in part by US NSF through grants IIS-0905215, CNS-1115234, IIS-0914934, DBI-0960443, and OISE-1129076, US Department of Army through grant W911NF-12-1-0066, Google Mobile 2014 Program and KAU grant, Natural Science Foundation of China (61070033, 61203280, 61202270), Natural Science Foundation of Guangdong province (9251009001000005, S2011040004187, S2012040007078), Specialized Research Fund for the Doctoral Program of Higher Education (20124420120004), Australian Research Council Discovery Grant (DP1096218, DP130102691) and ARC Linkage Grant (LP100200774, LP120100566).

References

1. Aggarwal, C.C., Yu, P.S.: A framework for clustering uncertain data streams. In: ICDE, pp. 150–159 (2008)
2. Aggarwal, C.C., Yu, P.S.: A survey of uncertain data algorithms and applications. TKDE 21(5), 609–623 (2009)
3. Cao, B., Pan, J., Zhang, Y., Yeung, D.Y., Yang, Q.: Adaptive transfer learning. In: AAAI (2010)
4. Fung, G.P.C., Yu, J.X., Lu, H., Yu, P.S.: Text classification without negative examples revisit. TKDE 18(6), 6–20 (2006)
5. Hido, S., Tsuboi, Y., Kashima, H., Sugiyama, M., Kanamori, T.: Statistical outlier detection using direct density ratio estimation. KAIS 26(2), 309–336 (2011)
6. Huffel, S.V., Vandewalle, J.: The total least squares problem: Computational aspects and analysis. Frontiers in Applied Mathematics, vol. 9. SIAM Press, Philadelphia (1991)

7. Yang, J., Gunn, S.: Exploiting uncertain data in support vector classification. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007/ WIRN 2007, Part III. LNCS (LNAI), vol. 4694, pp. 148–155. Springer, Heidelberg (2007)
8. Kao, B., Lee, S.D., Lee, F.K.F., Cheung, D.W., Ho, W.: Clustering uncertain data using voronoi diagrams and r-tree index. TKDE 22(9), 1219–1233 (2010)
9. Kriegel, H.P., Pfeifle, M.: Hierarchical density based clustering of uncertain data. In: ICDE, pp. 689–692 (2005)
10. Lawrence, N.D., Platt, J.C.: Learning to learn with the informative vector machine. In: ICML (2004)
11. Li, J., Su, L., Cheng, C.: Finding pre-images via evolution strategies. Applied Soft Computing 11(6), 4183–4194 (2011)
12. Li, X., Liu, B.: Learning to classify texts using positive and unlabeled data. In: IJCAI, pp. 587–592 (2003)
13. Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S.: Building text classifiers using positive and unlabeled examples. In: ICDM, pp. 179–186 (2003)
14. Liu, B., Xiao, Y., Cao, L., Yu, P.S.: One-class-based uncertain data stream learning. In: SDM, pp. 992–1003 (2011)
15. Pan, S.J., Tsand, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE TNN 22(2), 199–210 (2011)
16. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.Y.: Self-taught learning: transfer learning from unlabeled data. In: ICML (2007)
17. Schölkopf, B., Platt, J., Taylor, J.S., Smola, A.J., Williamson, R.: Estimating the support of a high-dimensional distribution. Neural Computation 13, 1443–1471 (2001)
18. Takruri, M., Rajasegarar, S., Challa, S., Leckie, C., Palaniswami, M.: Spatio-temporal modelling-based drift-aware wireless sensor networks. Wireless Sensor Systems 1(2), 110–122 (2011)
19. Tax, D.M.J., Duin, R.P.W.: Support vector data description. Machine Learning 54(1), 45–66 (2004)
20. TEvgeniou, T., Pontil, M.: Regularized multi–task learning. In: KDD (2004)
21. Trung, L., Dat, T., Phuoc, N., Wanli, M., Sharma, D.: Multiple distribution data description learning method for novelty detection. In: IJCNN, pp. 2321–2326 (2011)
22. Vapnik, V.: Statistical learning theory. Springer, London (1998)
23. William, J., Shaw, M.: On the foundation of evaluation. American Society for Information Science 37(5), 346–348 (1986)
24. Xiao, Y., Liu, B., Yin, J., Cao, L., Zhang, C., Hao, Z.: Similarity-based approach for positive and unlabeled learning. In: IJCAI, pp. 1577–1582 (2011)
25. Yu, H., Han, J., Chang, K.C.C.: Pebl: Web page classification without negative examples. TKDE 16(1), 70–81 (2004)