Knowl Inf Syst
DOI 10.1007/s10115-014-0765-8

REGULAR PAPER

A robust one-class transfer learning method with uncertain data

Yanshan Xiao · Bo Liu · Philip S. Yu · Zhifeng Hao

Received: 14 July 2013 / Revised: 10 May 2014 / Accepted: 3 July 2014

Abstract One-class classification aims at constructing a distinctive classifier based on one class of examples. Most of the existing one-class classification methods are proposed based on the assumptions that: (1) there are a large number of training examples available for learning the classifier; (2) the training examples can be explicitly collected and hence do not contain any uncertain information. However, in real-world applications, these assumptions are not always satisfied. In this paper, we propose a novel approach called uncertain one-class transfer learning with support vector machine (UOCT-SVM), which is capable of constructing an accurate classifier on the target task by transferring knowledge from multiple source tasks whose data may contain uncertain information. In UOCT-SVM, the optimization function is formulated to deal with uncertain data and transfer learning based on one-class SVM. Then, an iterative framework is proposed to solve the optimization function. Extensive experiments have showed that UOCT-SVM can mitigate the effect of uncertain data on the decision boundary and transfer knowledge from source tasks to help build an accurate classifier on the target task, compared with state-of-the-art one-class classification methods.

Keywords Transfer learning · Uncertain data · One-class classification

Y. Xiao · Z. Hao
School of Computers, Guangdong University of Technology, Guangzhou, China
e-mail: xiaoyanshan@gmail.com

Z. Hao
e-mail: mazfhao@scut.edu.cn

B. Liu (✉)
School of Automation, Guangdong University of Technology, Guangzhou, China
e-mail: csbliu@gmail.com

P. S. Yu
Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA
e-mail: psyu@uic.edu

P. S. Yu
Computer Science Department, King Abdulaziz University, Jeddah, Saudi Arabia

Published online: 29 July 2014

1 Introduction

One-class classification [1,2] is an important research area in machine learning and data mining, which addresses the learning problems where only one class of examples is available for training the classifier. In this case, the class available to learn the classifier is called the *target class*, and the other classes are the *nontarget classes*. The main task of one-class classification is to build a classifier on the target class, and the obtained classifier is then used to classify a new example to the target class or nontarget class [3]. One-class classification has found a large variety of applications, such as anomaly detection [1], automatic image annotation [4], sensor data drift detection [5] and remote sensing [6].

Depending on the availability of training data, the existing work on one-class classification can be classified into two broad categories. (1) The approaches for one-class classification with positive data and unlabeled data [7,8], where different algorithms are proposed to extract the negative examples from the unlabeled data, and then, a binary classifier is constructed based on the positive examples and the extracted negative examples. For example, Yu et al. [7] extracts negative documents by checking the frequency of features within the positive and unlabeled training documents and trains SVM iteratively to build a binary classifier. (2) The approaches for one-class classification with positive examples [1,9], where the classifier is trained on only the positive examples. For example, one-class SVM [1] is proposed to construct a hyperplane for separating the positive examples from the origin, such that the hyperplane can be utilized to differentiate the outliers (negative examples) from the positive examples. Our approach belongs to the second category.

Despite much progress on one-class classification, most of the existing work considers the one-class classification problem 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 [10]. Taking Web document classification as an example, the user is interested in football and plenty of Web documents are labeled on it. As time goes by, the user changes his interest to basket ball, which has related, but different data distributions from football. For this new task, we may not have many labeled documents since labeling a large number of documents timely may be expensive for the user. Hence, we expect to transfer the knowledge from previously labeled documents to help build the classifier of the new task. Another important observation is that, due to sampling error or instrument imperfection, the collected data in real-world applications may be corrupted with noises and thereafter contain uncertain information [11]. For example, in environmental monitoring applications, sensor networks generate a large amount of uncertain data because of instrument errors, limited accuracy or noise-prone wireless transmission [11]. Hence, how to build a classifier on the target task by transferring knowledge from the source task whose input data may contain uncertain information remains a key challenge for real-world one-class applications.

In this paper, we address the one-class transfer learning problem with uncertain data. To build the classifier, we have two challenges. The first one is how to construct the one-class classifier when the training examples on the target task are not sufficient to build a precise classifier and may be corrupted by noises. The second one is how to solve the formulated optimization problem effectively. To handle these challenges, we propose a novel approach, called uncertain one-class transfer learning with support vector machine (UOCT-SVM). The main characteristics of our approach can be viewed from the following aspects:

1. We propose an uncertain one-class transfer learning classifier, which can improve the one-class classifier of the target task by transferring knowledge from source tasks which

may contain uncertain data. At the same time, it can fulfill the knowledge transferring from not only a single source task but also multiple source tasks.
2. We propose an iterative framework to mitigate the effect of noises 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 one-class classification.
3. We conduct extensive experiments to evaluate the performance of our UOCT-SVM approach. The experimental results show that UOCT-SVM learns a more accurate classifier for the target task by transferring the knowledge from the source task and meanwhile mitigating the noises of the input data, compared with state-of-the-art one-class classification methods.

The rest of this paper is organized as follows. Section 2 discusses the existing work related to our study. Section 3 introduces the preliminaries. Section 4 presents our proposed approach in details. Section 5 extends our approach to knowledge transfer from multiple source tasks. Section 6 reports the experimental results. Section 7 concludes the paper and offers the future work.

2 Related work

2.1 Mining uncertain data

To deal with data uncertainty, many learning algorithms have been proposed. In the following, we briefly review the work on mining uncertain data in classification, clustering and other problems.

Some methods are proposed to deal with uncertain data in clustering problems. Kriegel and Pfeifle [12] adopts a fuzzy distance function to measure the similarity between uncertain data on top of the hierarchical density-based clustering algorithm. Ngai et al. [13] studies the problem of clustering data objects whose locations are uncertain and applies the UK-means algorithm to cluster uncertain objects. Aggarwal [14] discusses how to use the density-based approaches to handle error-prone and missing data.

Some other methods are devised to handle uncertain data in classification problems. Bi and Zhang [15] extends standard SVM to deal with uncertain data, which provides a geometric algorithm by optimizing the probabilistic separation between the two classes on both sides of the boundary. Gao and Wang [16] mines discriminative patterns from uncertain data as classification features/rules, to help train either SVM or rule-based classifier. Tsang et al. [17] modifies classical decision tree building algorithms to handle data tuples with uncertain values.

Recently, Murthy et al. [18] describes how aggregation is handled in the Trio system for uncertain and probabilistic data. Yuen et al. [19] proposes a new problem, called superseding nearest neighbor search, on uncertain spatial databases. Sun et al. [20] studies the discovery of frequent patterns and association rules from probabilistic data under the possible world semantics and proposes two efficient algorithms to discover frequent patterns in bottom-up and top-down manners.

Despite much progress on this area, most of the existing work considers uncertain data mining as a single task learning problem. However, in real-world applications, it may be expensive and time-consuming to label a large amount of data for a new learning task, and we expect to reduce the labeling efforts of the new task by transferring knowledge from

related tasks. In this paper, we propose a novel approach UOCT-SVM that can not only handle uncertain data but also improve classification accuracy of the new task's classifier by transferring knowledge from related tasks.

2.2 Transfer learning

In transfer learning, algorithms are designed to transfer knowledge to the target task from one or more source tasks that have similar, but not the same data distributions to the target task, such that the knowledge from the source task can benefit the learned classifier for the target task [10,21]. Transfer learning has been applied to solve the learning problems in various areas, such as text categorization [22–24], WiFi localization [25,26] and computer aided design [27].

According to Pan and Qiang [28], the approaches for transfer learning can be broadly categorized into four categories: instance-transfer, feature-representation-transfer, parameter-transfer, and relational-knowledge-transfer. In instance-transfer [21–24,29], training instances in the source domain are re-weighted according to their impact on the learning in the target task. In feature-representation-transfer [30–32], different algorithms are proposed to learn a common feature representation across tasks that reduces the task divergence and training error. In parameter-transfer [33–35], they attempt to discover the shared parameters between the source task and target task, which benefits for transfer learning. In relational-knowledge-transfer [36–38], the source task and target task are assumed to be relational, and the mapping of relational knowledge between the source task and target task is built. Our approach falls into the parameter-transfer category.

Multitask learning [25] is closely related to transfer learning. In multitask learning, multiple related tasks are learnt simultaneously to improve the predictive performance relative to learning these tasks independently. Bonilla et al. [39] investigates the multitask learning problems when task-specific features are available. They consider the similarity between tasks and construct a free-form kernel matrix to represent task relations. Lawrence and Platt [33] extends the informative vector machine to handle multitask learning cases. Yu et al. [40] proposes a hierarchical Gaussian process framework for multitask learning. Though multitask learning is related to transfer learning, they focus on different learning objectives. Multitask learning tries to improve the performances on all tasks, while transfer learning attempts to transfer knowledge from the source task to the target task [41].

Most of the existing work on transfer learning assumes that the training data in the source task and the target task can be precisely collected and does not contain any uncertain information. However, in real-world applications, due to sampling error or instrument imperfection, the collected data may be corrupted with noises and contain uncertain information. In this case, how to train a classifier that can fulfill the knowledge transferring, and meanwhile, deal with data uncertainty effectively becomes a key challenge for real-world transfer learning applications. This motivates the work in this paper.

3 Preliminary

3.1 One-class SVM

Suppose that the training target class is $S = \{\mathbf{x}_{11}, \mathbf{x}_{12}, \ldots, \mathbf{x}_{1|S|}\}$, where $\mathbf{x}_{1i} \in R^d$ is the ith example in S. One-class SVM aims to determine a plane to separate the target class and the origin of the space:

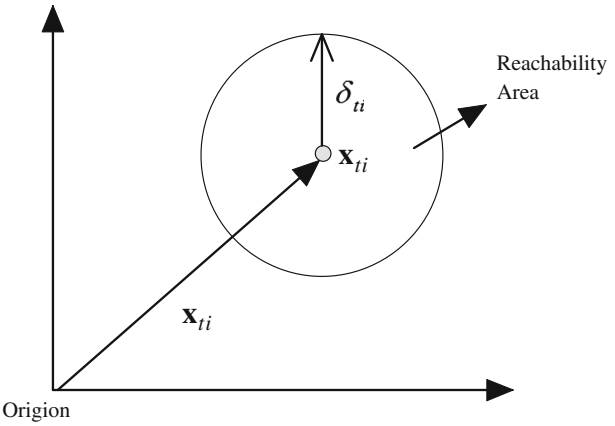


Fig. 1 Illustration of the reachability area of example $\mathbf{x}_{ti}$

$$\begin{aligned} \min \quad & \frac{1}{2} \parallel \mathbf{w}_0 \parallel^2 -\rho + C \sum_{i=1}^{|S|} \xi_i \\ \text{s.t.} \quad & \mathbf{w}_0^T \mathbf{x}_{1i} \geq \rho - \xi_i \\ & \xi_i \geq 0, \quad i = 1, 2, \ldots, |S|, \end{aligned} \tag{1}$$

where $\mathbf{w}_0$ and ρ are the norm vector and bias, respectively; C is a parameter trading off the margin and the errors. After solving problem (1), we can obtain the classifier $f(\mathbf{x}) = \mathbf{w}_0^T \mathbf{x} - \rho$. For a test example $\mathbf{x}$, if it has $\mathbf{w}_0^T \mathbf{x} - \rho > 0$, it is classified into the target class; otherwise, it belongs to the nontarget class.

In this paper, we extend standard one-class SVM to one-class transfer learning with uncertain data.

3.2 Uncertain data model

Since the collected example may deviate from the location that it should be, we assume that the example is subject to an additive noise vector $\Delta\mathbf{x}$. The original uncorrupted input $\mathbf{x}^s$ can thereafter be denoted as

$$\mathbf{x}^s = \mathbf{x} + \Delta\mathbf{x}. \tag{2}$$

In practice, we may not have any prior knowledge about the distribution of $\Delta\mathbf{x}$. For this reason, we assume that the noise vector $\Delta\mathbf{x}$ follows a particular distribution. The method of bounded and ellipsoidal uncertainties has been widely investigated and successfully applied in machine learning problems [9,42]. As in Liu et al. [9] and Huffel and Vandewalle [42], we consider a simple bound score δ for each example as

$$\parallel \Delta\mathbf{x} \parallel \leq \delta, \tag{3}$$

where $\parallel \Delta\mathbf{x} \parallel$ represents the norm of example $\Delta\mathbf{x}$. It is seen from (3) that the norm of $\Delta\mathbf{x}$ is no less than a bound score δ. We let $\mathbf{x} + \Delta\mathbf{x}$ ($\parallel \Delta\mathbf{x} \parallel \leq \delta$) denote the *reachability area* of example $\mathbf{x}$, as illustrated in Fig. 1. Then, it has

$$\parallel \mathbf{x}^s \parallel = \parallel \mathbf{x} + \Delta\mathbf{x} \parallel \leq \parallel \mathbf{x} \parallel + \parallel \Delta\mathbf{x} \parallel \leq \parallel \mathbf{x} \parallel + \delta \tag{4}$$

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

4 One-class transfer learning on uncertain data

In this section, we will present our proposed UOCT-SVM approach to handle uncertain data in one-class transfer learning when a single source task is available. The extension to transfer learning from more than one source task will be discussed in Sect. 5. Then, we will show how to solve the learning problem.

4.1 Formulation

Suppose that there are two one-class classification tasks—the source task S_1 and the target task S_2. The source task S_1 consists of $|S_1|$ positive examples, i.e., $\mathbf{x}_{11}, \mathbf{x}_{12}, \ldots, \mathbf{x}_{1|S_1|}$, where $\mathbf{x}_{1i} (i = 1, \ldots, |S_1|)$ denotes the ith examples in the source task S_1, and $|S_1|$ is the number of examples in S_1. Likewise, the target task S_2 contains $|S_2|$ positive examples, i.e., $\mathbf{x}_{21}, \mathbf{x}_{22}, \ldots, \mathbf{x}_{2|S_2|}$, where $|S_2|$ is the number of examples in S_2. The main objective of one-class transfer learning is to transfer the knowledge of the source task S_1 to the target task S_2. Let $f_1(\mathbf{x}) = \mathbf{w}_1^T\mathbf{x} - \rho_1$ and $f_2(\mathbf{x}) = \mathbf{w}_2^T\mathbf{x} - \rho_2$ be the classification planes for S_1 and S_2, respectively. To facilitate the transfer, we make

$$\mathbf{w}_1 = \mathbf{w}_0 + \mathbf{v}_1, \tag{5}$$

$$\mathbf{w}_2 = \mathbf{w}_0 + \mathbf{v}_2, \tag{6}$$

where $\mathbf{w}_0$ can be considered as a bridge to transfer knowledge from the source task to the target task; $\mathbf{v}_1$ and $\mathbf{v}_2$ represent the discrepancy between the globe optimal decision boundary ($\mathbf{w}_0$) and the local optimal decision boundary ($\mathbf{w}_0 + \mathbf{v}_1$ for the source task S_1 and $\mathbf{w}_0 + \mathbf{v}_2$ for the target task S_2). By substituting Eqs. (5) and (6), the hyperplanes for S_1 and S_2 can be rewritten as $f_1(\mathbf{x}) = (\mathbf{w}_0 + \mathbf{v}_1)^T\mathbf{x} - \rho_1$ and $f_2(\mathbf{x}) = (\mathbf{w}_0 + \mathbf{v}_2)^T\mathbf{x} - \rho_2$. Moreover, the input examples in the source task and the target task may contain uncertain information. To deal with the uncertainty, we represent each example of S_1 and S_2 as $\mathbf{x} + \Delta\mathbf{x}$, based on the uncertain data model in Sect. 3.2. Hence, the learning problem for one-class transfer learning with uncertain data can be formulated as

$$\begin{aligned}
\min \quad & \|\mathbf{w}_0\|^2 + C_1\|\mathbf{v}_1\|^2 + C_2\|\mathbf{v}_2\|^2 - \rho_1 - \rho_2 + \sum_{t=1}^{2} C_t \sum_{i=1}^{|S_t|} \xi_{ti} \\
\text{s.t.} \quad & (\mathbf{w}_0 + \mathbf{v}_1)^T(\mathbf{x}_{1i} + \Delta\mathbf{x}_{1i}) \geq \rho_1 - \xi_{1i}, \quad i = 1, \ldots, |S_1| \\
& (\mathbf{w}_0 + \mathbf{v}_2)^T(\mathbf{x}_{2j} + \Delta\mathbf{x}_{2j}) \geq \rho_2 - \xi_{2j}, \quad j = 1, \ldots, |S_2| \\
& \xi_{1i} \geq 0, \quad \xi_{2j} \geq 0, \quad \|\Delta\mathbf{x}_{1i}\| \leq \delta_{1i}, \quad \|\Delta\mathbf{x}_{2j}\| \leq \delta_{2j}.
\end{aligned} \tag{7}$$

where C_1, C_2 and C_t are regularized parameters; ξ_{1i} and ξ_{2j} are training errors. From the optimization problem (7), we can observe that:

1 $\mathbf{x}_{1i}$ and $\mathbf{x}_{2j}$ are the imputed training examples, which may be corrupted by noises and contain uncertain information. To reduce the effect of noises, we let each example in the source task and the target task represented as $\mathbf{x}_{1i} + \Delta\mathbf{x}_{1i}$ and $\mathbf{x}_{2j} + \Delta\mathbf{x}_{2j}$, respectively.

By optimizing the values of $\Delta\mathbf{x}_{1i}$ and $\Delta\mathbf{x}_{2j}$, we can refine the learnt one-class transfer learning classifier less sensitive to noises. $\|\Delta\mathbf{x}_{1i}\| \leq \delta_{1i}$ and $\|\Delta\mathbf{x}_{2j}\| \leq \delta_{2j}$ restrict the range of uncertain information using bound scores δ_{1i} and δ_{2j}.

2 We utilize the common variable $\mathbf{w}_0$ as a bridge to transfer the knowledge. 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.

4.2 Solution to uncertain one-class transfer learning classifier

The optimization function (7) is difficult to solve since the variables $\mathbf{w}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$, ρ_1, ρ_2, $\Delta\mathbf{x}_{1i}$, $\Delta\mathbf{x}_{2j}$, ξ_{1i} and ξ_{2j} are unknown to us. In this section, we will employ an iterative framework to calculate these unknown variables and present a novel scheme to estimate the bound score δ_{1i} and δ_{2j} for the training examples.

Specifically, the iterative framework consists of two steps. In the first step, we fix $\Delta\mathbf{x}_{1i}$ and $\Delta\mathbf{x}_{2j}$, and solve the learning problem (7) to obtain the values of $\mathbf{w}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$, ρ_1, ρ_2, ξ_{1i} and ξ_{2j}. In the second step, we fix $\mathbf{w}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$, ρ_1, ρ_2, ξ_{1i} and ξ_{2j}, and optimize the learning problem (7) to get the values of $\Delta\mathbf{x}_{1i}$ and $\Delta\mathbf{x}_{2j}$. The above two steps repeat alternatively until the termination criterion is met. In the following, we present the two steps in details.

4.2.1 Calculating the classifier by fixing $\Delta\boldsymbol{x}_{1i}$ and $\Delta\boldsymbol{x}_{2j}$

We initialize $\Delta\mathbf{x}_{1i}$ and $\Delta\mathbf{x}_{2j}$ as $\Delta\bar{\mathbf{x}}_{1i}$ and $\Delta\bar{\mathbf{x}}_{2j}$, respectively, and let them satisfy the constraints $\| \Delta\bar{\mathbf{x}}_{1i} \| \leq \delta_{1i}$ and $\| \Delta\bar{\mathbf{x}}_{2j} \| \leq \delta_{2j}$.[1] Then, the constraints $\| \Delta\bar{\mathbf{x}}_{1i} \| \leq \delta_{1i}$, $\| \Delta\bar{\mathbf{x}}_{2j} \| \leq \delta_{2j}$ in problem (7) will not have effect on the solution, and we remove them from the objective function. The objective function (7) is transformed into

$$\begin{aligned} \min \quad & \|\mathbf{w}_0\|^2 + C_1\|\mathbf{v}_1\|^2 + C_2\|\mathbf{v}_2\|^2 - \rho_1 - \rho_2 + \sum_{t=1}^{2} C_t \sum_{i=1}^{|S_t|} \xi_{ti} \\ \text{s.t.} \quad & (\mathbf{w}_0 + \mathbf{v}_1)^T (\mathbf{x}_{1i} + \Delta\bar{\mathbf{x}}_{1i}) \geq \rho_1 - \xi_{1i}, \quad i = 1, \ldots, |S_1| \\ & (\mathbf{w}_0 + \mathbf{v}_2)^T (\mathbf{x}_{2j} + \Delta\bar{\mathbf{x}}_{2j}) \geq \rho_2 - \xi_{2j}, \quad j = 1, \ldots, |S_2| \\ & \xi_{1i} \geq 0, \quad \xi_{2j} \geq 0. \end{aligned} \tag{8}$$

Since the values of $\Delta\bar{\mathbf{x}}_{1i}$ and $\Delta\bar{\mathbf{x}}_{2j}$ are given, problem (8) is a QP problem, which can be transformed into a standard one-class SVM and solved via the dual form. Hence, we give the dual form of problem (8) in Theorem 1.

Theorem 1 *By introducing the Lagrange function* [43]*, the dual form of the optimization problem* (8) *can be given by*

$$\begin{aligned} \min \quad & \frac{1}{2} \sum_{i=1}^{|S_1|} \sum_{j=1}^{|S_2|} \alpha_{1i} \bar{\mathbf{x}}_{1i}^T \bar{\mathbf{x}}_{2j} \alpha_{2j} + \frac{C_1 + 1}{4C_1} \sum_{h=1}^{|S_1|} \sum_{g=1}^{|S_1|} \alpha_{1h} \bar{\mathbf{x}}_{1h}^T \bar{\mathbf{x}}_{1g} \alpha_{1g} \\ & + \frac{C_2 + 1}{4C_2} \sum_{p=1}^{|S_2|} \sum_{k=1}^{|S_2|} \alpha_{2p} \bar{\mathbf{x}}_{2p}^T \bar{\mathbf{x}}_{2k} \alpha_{2k} \end{aligned}$$

[1] In the experiments, we initialize $\Delta\bar{\mathbf{x}}_{1i} = 0$ and $\Delta\bar{\mathbf{x}}_{2j} = 0$.

$$\text{s.t.} \sum_{i=1}^{|S_1|} \alpha_{1i} = 1, \quad 0 \le \alpha_{1i} \le C_1, \quad i = 1, \ldots, |S_1|$$
$$\sum_{j=1}^{|S_2|} \alpha_{2j} = 1, \quad 0 \le \alpha_{2j} \le C_2, \quad j = 1, \ldots, |S_2| \tag{9}$$

where $\alpha_{1i}, \alpha_{2j}, \alpha_{1h}, \alpha_{1g}, \alpha_{2p}, \alpha_{2k} \ge 0$ *are Lagrange multipliers; it has* $\overline{\boldsymbol{x}}_{1i} = \boldsymbol{x}_{1i} + \Delta\overline{\boldsymbol{x}}_{1i}$*;* $\overline{\boldsymbol{x}}_{2j}, \overline{\boldsymbol{x}}_{1g}, \overline{\boldsymbol{x}}_{1h}, \overline{\boldsymbol{x}}_{2p}$ *and* $\overline{\boldsymbol{x}}_{2k}$ *are similar to* $\overline{\boldsymbol{x}}_{1i}$.

The proof for obtaining Theorem 1 can refer to Sect. 8.1. After solving the dual form (9), we can obtain the solutions of $\alpha_{1i}, \alpha_{2j}, \alpha_{1h}, \alpha_{1g}, \alpha_{2p}$, and α_{2k}. Then, the values of $\mathbf{w}_0$, $\mathbf{v}_1$ and $\mathbf{v}_2$ can be calculated as

$$\mathbf{w}_0 = \frac{1}{2}\left(\sum_{i=1}^{|S_1|} \alpha_{1i}\overline{\mathbf{x}}_{1i} + \sum_{j=1}^{|S_2|} \alpha_{2j}\overline{\mathbf{x}}_{2j}\right), \tag{10}$$

$$\mathbf{v}_1 = \frac{1}{2C_1}\sum_{i=1}^{|S_1|} \alpha_{1i}\overline{\mathbf{x}}_{1i}, \tag{11}$$

$$\mathbf{v}_2 = \frac{1}{2C_2}\sum_{j=1}^{|S_2|} \alpha_{2j}\overline{\mathbf{x}}_{2j}, \tag{12}$$

For the examples $\mathbf{x}_{1i}$ in S_1, we let subset S_1^* contain those examples with $0 < \alpha_{1i} < C_1$. For the examples $\mathbf{x}_{2j}$ in S_2, we let subset S_2^* contain those examples with $0 < \alpha_{2j} < C_2$. According to the KKT conditions [43], the examples with $0 < \alpha_{ti} < C_t$ $(i = 1, \ldots, |S_t|, t = 1, 2)$ are support vectors (SVs) whose corresponding constraints become equation and it has $\xi_{ti} = 0$. Hence, we obtain

$$(\mathbf{w}_0 + \mathbf{v}_1)^T\overline{\mathbf{x}}_{1i} = \rho_1, \quad \mathbf{x}_{1i} \in S_1^* \tag{13}$$
$$(\mathbf{w}_0 + \mathbf{v}_2)^T\overline{\mathbf{x}}_{2j} = \rho_2, \quad \mathbf{x}_{2j} \in S_2^* \tag{14}$$

According to Eqs. (13) and (14), ρ_1 and ρ_2 can be computed as

$$\rho_1 = \frac{1}{|S_1^*|}\sum_{\mathbf{x}_{1i} \in S_1^*} (\mathbf{w}_0 + \mathbf{v}_1)^T\overline{\mathbf{x}}_{1i}, \tag{15}$$

$$\rho_2 = \frac{1}{|S_2^*|}\sum_{\mathbf{x}_{2j} \in S_2^*} (\mathbf{w}_0 + \mathbf{v}_2)^T\overline{\mathbf{x}}_{2j}, \tag{16}$$

where $|S_1^*|$ and $|S_2^*|$ represent the corresponding numbers of examples in S_1^* and S_2^*. Based on the values of $\mathbf{w}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$, ρ_1 and ρ_2, the classifier for the target task can be obtained as $f_2(\mathbf{x}) = (\mathbf{w}_0 + \mathbf{v}_2)^T\mathbf{x} - \rho_2$.

4.2.2 Calculating $\Delta\boldsymbol{x}_{1i}$ *and* $\Delta\boldsymbol{x}_{2j}$ *by fixing the classifier*

Supposing that $f_1(\mathbf{x}) = (\overline{\mathbf{w}}_0 + \overline{\mathbf{v}}_1)^T\mathbf{x} - \overline{\rho}_1$ and $f_2(\mathbf{x}) = (\overline{\mathbf{w}}_0 + \overline{\mathbf{v}}_2)^T\mathbf{x} - \overline{\rho}_2$ are the classifier obtained from the first step. Here, we fix the classifiers $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$, i.e., making $\mathbf{w}_0 = \overline{\mathbf{w}}_0$, $\mathbf{v}_1 = \overline{\mathbf{v}}_1$, $\mathbf{v}_2 = \overline{\mathbf{v}}_2$, $\rho_1 = \overline{\rho}_1$ and $\rho_2 = \overline{\rho}_2$ and optimize problem (7) over $\Delta\mathbf{x}_{1i}$ and $\Delta\mathbf{x}_{2j}$. To do this, we have Theorem 2 as follows.

Algorithm 1 Uncertain one-class transfer learning with uncertain data

Input: Source task S_1, target task S_2 ; // Training set
..... C_1, C_2; // Regularization parameters
..... δ_{1i}, δ_{2j}; // bound scores for training examples.
Output: $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$.
1: t=0;
2: Initialize $F_{val}(t) = \infty$;
3: **repeat**
4: $t = t + 1$;
5: **if** t=1 **then**
6: Initialize $\Delta\bar{\mathbf{x}}_{1i} = 0$ and $\Delta\bar{\mathbf{x}}_{2j} = 0$;
7: **else**
8: Update $\Delta\bar{\mathbf{x}}_{1i}$ and $\Delta\bar{\mathbf{x}}_{2j}$ based on (17) and (18), by fixing $\mathbf{w}_0, \mathbf{v}_1, \mathbf{v}_2, \rho_1, \rho_2$;
9: **end if**
10: Substitute $\Delta\bar{\mathbf{x}}_{1i}$ and $\Delta\bar{\mathbf{x}}_{2j}$, and solve problem (8);
11: Compute $\mathbf{w}_0$, $\mathbf{v}_1$ and $\mathbf{v}_2$ according to Equations (10)-(12);
12: Compute ρ_1 and ρ_2 based on Equations (15)-(16);
13: Let $F_{val}(t)$ be the decision function's value of problem (8);
14: Let $F_{max} = \max\{|F_{val}(t-1)|, |F_{val}(t)|\}$
15: **until** $|F_{val}(t) - F_{val}(t-1)| < \varepsilon F_{max}$
16: Return $f_1(\mathbf{x}) = (\mathbf{w}_0 + \mathbf{v}_1)^T\mathbf{x} - \rho_1$ and $f_2(\mathbf{x}) = (\mathbf{w}_0 + \mathbf{v}_2)^T\mathbf{x} - \rho_2$.

Theorem 2 *By fixing $\boldsymbol{w}_0, \boldsymbol{v}_1, \boldsymbol{v}_2, \rho_1$ and ρ_2 to be $\overline{\boldsymbol{w}}_0, \overline{\boldsymbol{v}}_1, \overline{\boldsymbol{v}}_2, \overline{\rho}_1$ and $\overline{\rho}_2$, respectively, the solutions of $\Delta\boldsymbol{x}_{1i}$ and $\Delta\boldsymbol{x}_{2j}$ for optimizing problem (7) are*

$$\Delta\boldsymbol{x}_{1i} = \delta_{1i}\frac{\overline{\boldsymbol{w}}_0 + \overline{\boldsymbol{v}}_1}{\| \overline{\boldsymbol{w}}_0 + \overline{\boldsymbol{v}}_1 \|}, \quad i = 1, \ldots, |S_1|, \tag{17}$$

$$\Delta\boldsymbol{x}_{2j} = \delta_{2j}\frac{\overline{\boldsymbol{w}}_0 + \overline{\boldsymbol{v}}_2}{\| \overline{\boldsymbol{w}}_0 + \overline{\boldsymbol{v}}_2 \|}, \quad j = 1, \ldots, |S_2|. \tag{18}$$

It is seen from (7) that the objective function's value is determined by $\mathbf{w}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$, ρ_1, ρ_2 and $\sum_{t=1}^{2}\sum_{i=1}^{|S_t|}\xi_{ti}$. Considering that $\mathbf{w}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$, ρ_1 and ρ_2 are fixed to be $\overline{\mathbf{w}}_0$, $\overline{\mathbf{v}}_1$, $\overline{\mathbf{v}}_2$, $\overline{\rho}_1$ and $\overline{\rho}_2$, respectively, the objective function's value is decided by $\sum_{t=1}^{2}\sum_{i=1}^{|S_t|}\xi_{ti}$, and the optimization of the objective function (7) is transformed into the minimization of $\sum_{t=1}^{2}\sum_{i=1}^{|S_t|}\xi_{ti}$. In Theorem 2, we try to optimize $\Delta\mathbf{x}_{ti}$ and $\Delta\mathbf{x}_{tj}$ to minimize the value of $\sum_{t=1}^{2}\sum_{i=1}^{|S_t|}\xi_{ti}$. The proof for Theorem 2 refers to Sect. 8.2.

4.2.3 Iterative framework

We have introduced the details of how to train the classifiers $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ and update the noise vectors $\Delta\mathbf{x}_{1i}$ and $\Delta\mathbf{x}_{2j}$. By referring to the alternating optimization method in [42], an iterative framework is proposed to train the classifier and update the noise vectors alternatively until a termination criterion is met. Algorithm 1 illustrates the pseudo codes of our approach. Here, we employ the stopping criterion as in [44] to determine the termination of UOCT-SVM. When the proportion of $|F_{val}(t) - F_{val}(t-1)|$ and F_{max} is smaller than a threshold ε, the algorithm stops.

Moreover, in problems (7) and (8), δ_{1i} and δ_{2j} are parameters that we need to estimate. For each example $\mathbf{x}_{1i}$ in S_1, we calculate the average distance between $\mathbf{x}_{1i}$ and its k–nearest neighbors—and assign this average distance to δ_{1i}. The same operation is utilized to the examples $\mathbf{x}_{2j}$ in S_2. This setting has been successfully utilized in the previous work [9].

It is noted that the problem settings of our approach and "Transfer Learning with One-Class data" (TLOC) [45] are different. In our approach, the target task is a one-class classification problem, and the source tasks are also one-class classification problems. Our approach aims at describing the data distribution of the positive class. In TLOC, the target task is a one-class classification problem, but the source tasks are binary-class classification problems. TLOC attempts to depict the distributions of both the positive class and the negative class.

4.3 Kernelized uncertain one-class transfer learning classifier

In the nonlinear classification problems, the examples of the target class and the nontarget class are difficult to be separated by using a linear classification plane. To make the data more separable, we map the training examples into the feature space via a nonlinear mapping function $\phi(\cdot)$. Hence, the examples in the source task are transformed into $\{\phi(\mathbf{x}_{11}), \ldots, \phi(\mathbf{x}_{1|S_1|})\}$, and those in the target task are changed into $\{\phi(\mathbf{x}_{21}), \ldots, \phi(\mathbf{x}_{2|S_2|})\}$, where $\phi(\mathbf{x}_{ti})$ is the image of example $\mathbf{x}_{ti}$ in the feature space. The inner product of $\phi(\mathbf{x}_{ti})$ and $\phi(\mathbf{x}_{hj})$ can be calculated using a kernel function $K(\mathbf{x}_{ti}, \mathbf{x}_{hj}) = \phi(\mathbf{x}_{ti})^T \phi(\mathbf{x}_{hj})$.

To build the nonlinear classifier, we need to conduct some modifications on the two steps presented in Sect. 4.2. In the dual form (9) of the first step, we replace $\overline{\mathbf{x}}_{1i}^T \overline{\mathbf{x}}_{2j}$, $\overline{\mathbf{x}}_{1h}^T \overline{\mathbf{x}}_{1g}$ and $\overline{\mathbf{x}}_{2p}^T \overline{\mathbf{x}}_{2k}$ with $K(\overline{\mathbf{x}}_{1i}, \overline{\mathbf{x}}_{2j})$, $K(\overline{\mathbf{x}}_{1h}, \overline{\mathbf{x}}_{1g})$ and $K(\overline{\mathbf{x}}_{2p}, \overline{\mathbf{x}}_{2k})$, respectively. After solving the dual form, the corresponding classifiers for the source task and the target task are obtained as (19) and (20).

$$f_1(\phi(\mathbf{x})) = \frac{1}{2} \sum_{t=1}^{2} \sum_{i=1}^{|S_t|} \alpha_{ti} K(\overline{\mathbf{x}}_{ti}, \mathbf{x}) + \frac{1}{2C_1} \sum_{i=1}^{|S_1|} \alpha_{1i} K(\overline{\mathbf{x}}_{1i}, \mathbf{x}) - \rho_1 \tag{19}$$

$$f_2(\phi(\mathbf{x})) = \frac{1}{2} \sum_{h=1}^{2} \sum_{j=1}^{|S_h|} \alpha_{hj} K(\overline{\mathbf{x}}_{hj}, \mathbf{x}) + \frac{1}{2C_2} \sum_{j=1}^{|S_2|} \alpha_{2j} K(\overline{\mathbf{x}}_{2j}, \mathbf{x}) - \rho_2 \tag{20}$$

Moreover, in the input space, we estimate the uncertainties using bounded sphere as $\Delta\mathbf{x}_{ti} \leq \delta_{ti}$. However, in the feature space, the bounded spheres correspond to irregular shapes and it brings difficulty to solve the optimization problem. For this reason, we adopt an approximation strategy based on the first order Taylor expansion of the kernel function $K(\cdot)$. The first order Taylor expansion of $K(\cdot)$ with respect to $\mathbf{x}$ at point $\mathbf{x}_{ti}$ is

$$K(\mathbf{x}_{ti} + \Delta\mathbf{x}_{ti}, \cdot) = K(\mathbf{x}_{ti}, \cdot) + \Delta\mathbf{x}_{ti}^T K'(\mathbf{x}_{ti}, \cdot), \tag{21}$$

where $K'(\mathbf{x}_{ti}, \cdot)$ is the gradient of $K(\cdot)$ with respect to $\mathbf{x}$ at point $\mathbf{x}_{ti}$.

For the second step in Sect. 4.2, we minimize the value of $\sum_{t=1}^{2} \sum_{\mathbf{x}_{ti} \in S_t} \xi_{ti}$ over $\Delta\mathbf{x}_{1i}$ and $\Delta\mathbf{x}_{2j}$, which can be transformed into the minimization of each $\xi = \max\{0, \overline{\rho}_t - \frac{1}{2}\sum_{h=1}^{2}\sum_{j=1}^{|S_h|} \alpha_{hj} K(\mathbf{x}_{hj} + \Delta\overline{\mathbf{x}}_{hj}, \mathbf{x} + \Delta\mathbf{x}) - \frac{1}{2C_t}\sum_{j=1}^{|S_t|} \alpha_{tj} K(\mathbf{x}_{tj} + \Delta\overline{\mathbf{x}}_{tj}, \mathbf{x} + \Delta\mathbf{x}\}$ ($\mathbf{x} \in S_t, t = 1, 2$) over $\Delta\mathbf{x}$. By applying the first order Taylor expansion of $K(\cdot)$ in Eq. (21), we have Theorem 3 in the following.

Theorem 3 *Assuming that $\boldsymbol{w}_0, \boldsymbol{v}_1, \boldsymbol{v}_2, \rho_1$ and ρ_2 are fixed to $\overline{\boldsymbol{w}}_0, \overline{\boldsymbol{v}}_1, \overline{\boldsymbol{v}}_2, \overline{\rho}_1$ and $\overline{\rho}_2$, respectively, the optimal values of $\Delta\boldsymbol{x}_{1i}$ and $\Delta\boldsymbol{x}_{2j}$ are*

$$\Delta\mathbf{x}_{1i} = \delta_{1i} \frac{\mathbf{u}_{1i}}{||\mathbf{u}_{1i}||}, \quad \Delta\mathbf{x}_{2j} = \delta_{2j} \frac{\mathbf{u}_{2j}}{||\mathbf{u}_{2j}||}, \tag{22}$$

where it has

$$\boldsymbol{u}_{1i} = \frac{1}{2}\sum_{h=1}^{2}\sum_{g=1}^{|S_h|}\alpha_{hg}K'\left(\boldsymbol{x}_{hg}+\Delta\bar{\boldsymbol{x}}_{hg},\boldsymbol{x}_{1i}\right)+\frac{1}{2C_1}\sum_{g=1}^{|S_1|}\alpha_{1g}K'\left(\boldsymbol{x}_{1g}+\Delta\bar{\boldsymbol{x}}_{1g},\boldsymbol{x}_{1i}\right),$$

$$\boldsymbol{u}_{2j} = \frac{1}{2}\sum_{h=1}^{2}\sum_{g=1}^{|S_h|}\alpha_{hg}K'\left(\boldsymbol{x}_{hg}+\Delta\bar{\boldsymbol{x}}_{hg},\boldsymbol{x}_{2j}\right)+\frac{1}{2C_2}\sum_{g=1}^{|S_2|}\alpha_{2g}K'\left(\boldsymbol{x}_{2g}+\Delta\bar{\boldsymbol{x}}_{2g},\boldsymbol{x}_{2j}\right).$$

In Theorem 3, $K'(\boldsymbol{x}_{hg}+\Delta\bar{\boldsymbol{x}}_{hg},\boldsymbol{x}_{1i})$ and $K'(\boldsymbol{x}_{hg}+\Delta\bar{\boldsymbol{x}}_{hg},\boldsymbol{x}_{2j})$ are the gradient of $K(\boldsymbol{x}_{hg}+\Delta\bar{\boldsymbol{x}}_{hg},\boldsymbol{x})$ with respect to $\boldsymbol{x}$ at points $\boldsymbol{x}_{1i}$ and $\boldsymbol{x}_{2j}$, respectively. The proof for Theorem 3 can refer to Sect. 8.3. Similar to the linear cases, for the nonlinear classification problems, $\Delta\boldsymbol{x}_{1i}$ and $\Delta\boldsymbol{x}_{2j}$ are initialized to be zero vectors. In the first step, we fix $\Delta\boldsymbol{x}_{1i}$ and $\Delta\boldsymbol{x}_{2j}$ to obtain the nonlinear classifier by replacing $\bar{\boldsymbol{x}}_{1i}^T\bar{\boldsymbol{x}}_{2j}$, $\bar{\boldsymbol{x}}_{1h}^T\bar{\boldsymbol{x}}_{1g}$ and $\bar{\boldsymbol{x}}_{2p}^T\bar{\boldsymbol{x}}_{2k}$ with $K(\bar{\boldsymbol{x}}_{1i},\bar{\boldsymbol{x}}_{2j})$, $K(\bar{\boldsymbol{x}}_{1h},\bar{\boldsymbol{x}}_{1g})$ and $K(\bar{\boldsymbol{x}}_{2p},\bar{\boldsymbol{x}}_{2k})$, respectively, in the dual form (9). In the second step, we fix the classifier and optimize the value of $\Delta\boldsymbol{x}_{ti}$ as presented in (22). Since the initial values of $\Delta\boldsymbol{x}_{1i}$ and $\Delta\boldsymbol{x}_{2j}$ are given, $\Delta\boldsymbol{x}_{1i}$ and $\Delta\boldsymbol{x}_{2j}$ in the current iteration can be obtained based on their values in the previous iteration by using (22). The above two steps iterate until the termination criterion is met.

5 Extension to one-class transfer learning with uncertain data from multiple source tasks

In this section, we extend the uncertain one-class transfer learning classifier to the learning problems where the knowledge from more than one source tasks is transferred to the target task. Suppose that there are $K-1$ source tasks $S_1, S_2, \ldots, S_{K-1}$, and one target task S_K. Each task is one-class classification problem, which consists of a number of positive examples, i.e., $\mathbf{x}_{t1}, \mathbf{x}_{t2}, \ldots, \mathbf{x}_{t|S_t|}$, where $\mathbf{x}_{ti}$ $(i = 1, \ldots, |S_t|)$ is the ith example in task S_t, and $|S_t|$ is the number of examples in S_t. We aim at constructing an one-class transfer learning classifier that is capable of transferring the knowledge of the $K-1$ source tasks to the target task with uncertain data.

Let $f_i(\mathbf{x}) = (\mathbf{w}_0+\mathbf{v}_i)^T\mathbf{x}-\rho_i$ be the classification plane for the ith source tasks $(i = 1, \ldots, K-1)$, and $f_K(\mathbf{x}) = (\mathbf{w}_0+\mathbf{v}_K)^T\mathbf{x}-\rho_K$ denote the plane for the target task. Since e ach example in the source and target tasks may contain uncertain information, similar to Sect. 4, we introduce a noise vector $\Delta\mathbf{x}_{ti}$ for each example $\mathbf{x}_{ti}$, and the corrected example is represented as $\mathbf{x}_{ti}+\Delta\mathbf{x}_{ti}$. Based on this uncertain data representation, the learning problem of the uncertain one-class transfer learning classifier with multiple source tasks can be given as

$$\begin{aligned}
\min\ & ||\mathbf{w}_0||^2+\frac{1}{K}\sum_{t=1}^{K}C_t||\mathbf{v}_t||^2-\sum_{t=1}^{K}\rho_t+\sum_{t=1}^{K}C_t\sum_{i=1}^{|S_t|}\xi_{ti}\\
\text{s.t.}\ & (\mathbf{w}_0+\mathbf{v}_t)^T(\mathbf{x}_{ti}+\Delta\mathbf{x}_{ti})\geq\rho_t-\xi_{ti},\quad i=1,\ldots,|S_t|,\quad t=1,\ldots,K\\
& ||\Delta\mathbf{x}_{ti}||\leq\delta_{ti},\quad \xi_{ti}\geq 0.
\end{aligned}\tag{23}$$

It is seen from problem (23) that for the corrected examples $\mathbf{x}_{ti}+\Delta\mathbf{x}_{ti}$ in task S_t, the corresponding classifier $f_t(\mathbf{x}) = (\mathbf{w}_0+\mathbf{v}_t)^T\mathbf{x}-\rho_t$ separates them from the origin with a margin. $\mathbf{w}_0$ is considered as a common variable to transfer the knowledge between the source tasks and the target task.

Similar to Sect. 4, an iterative framework is adopted to solve problem (23). In the first step, we initialize $\Delta \mathbf{x}_{ti}$ to be 0, and fix them to train the uncertain one-class transfer learning classifier. Since the values of $\Delta \mathbf{x}_{ti}$ are no larger than δ_{ti}, we eliminate the constraints $||\Delta \mathbf{x}_{ti}|| \leq \delta_{ti}$, and problem (23) is changed into

$$\begin{aligned} \min \quad & ||\mathbf{w}_0||^2 + \frac{1}{K}\sum_{t=1}^{K} C_t ||\mathbf{v}_t||^2 - \sum_{t=1}^{K}\rho_t + \sum_{t=1}^{K} C_t \sum_{i=1}^{|S_t|}\xi_{ti} \\ \text{s.t.} \quad & (\mathbf{w}_0 + \mathbf{v}_t)^T(\mathbf{x}_{ti} + \Delta \mathbf{x}_{ti}) \geq \rho_t - \xi_{ti}, \quad i = 1, \ldots, |S_t|, \quad t = 1, \ldots, K \\ & \xi_{ti} \geq 0. \end{aligned} \tag{24}$$

To solve problem (24), we first let $\mathbf{e} = (1, 1, \ldots, 1)^T$ be a K-dimensional column vector, $\rho = (\rho_1, \rho_2, \ldots, \rho_K)^T$, and redefine the following notations.

$$\mathbf{w} = \left(\mathbf{w}_0, \sqrt{\frac{C_1}{K}}\mathbf{v}_1, \sqrt{\frac{C_2}{K}}\mathbf{v}_2, \ldots, \sqrt{\frac{C_K}{K}}\mathbf{v}_K\right)^T \tag{25}$$

$$\mathbf{z}(\mathbf{x}_{ti}, t) = \left(\mathbf{x}_{ti}, \underbrace{\mathbb{0}, \ldots, \mathbb{0}}_{t-1}, \sqrt{\frac{K}{C_t}}\mathbf{x}_{ti}, \underbrace{\mathbb{0}, \ldots, \mathbb{0}}_{K-t}\right)^T \tag{26}$$

$$\Delta\mathbf{z}(\mathbf{x}_{ti}, t) = \left(\Delta\mathbf{x}_{ti}, \underbrace{\mathbb{0}, \ldots, \mathbb{0}}_{t-1}, \sqrt{\frac{K}{C_t}}\Delta\mathbf{x}_{ti}, \underbrace{\mathbb{0}, \ldots, \mathbb{0}}_{K-t}\right)^T \tag{27}$$

$$\mathbf{e}_t = \left(\underbrace{0, \ldots, 0}_{t-1}, \mathbb{1}, \underbrace{0, \ldots, 0}_{K-t}\right)^T \tag{28}$$

where $\mathbb{0} = \{0, 0, \ldots, 0\}^T$ and $\mathbb{1} = \{1, 1, \ldots, 1\}^T$ are d-dimensional column vectors with all elements being 0 and 1, respectively. Based on the above notations, problem (24) can be transformed into

$$\begin{aligned} \min \quad & ||\mathbf{w}||^2 - \rho^T\mathbf{e} + \sum_{t=1}^{K} C_t \sum_{i=1}^{|S_t|}\xi_{ti} \\ \text{s.t.} \quad & \mathbf{w}^T\bar{\mathbf{z}}(\mathbf{x}_{ti}, t) \geq \rho^T\mathbf{e}_t - \xi_{ti}, \quad i = 1, \ldots, |S_t|, \quad t = 1, \ldots, K \\ & \xi_{ti} \geq 0. \end{aligned} \tag{29}$$

where it has $\bar{\mathbf{z}}(\mathbf{x}_{ti}, i) = \mathbf{z}(\mathbf{x}_{ti}, t) + \Delta\mathbf{z}(\mathbf{x}_{ti}, t)$. Problem (29) is a standard one-class SVM, which is solved via the dual from, as presented in Theorem 4.

Theorem 4 *The dual form of problem* (29) *can be obtained as*

$$\begin{aligned} \max \quad & -\frac{1}{4}\sum_{t=1}^{K}\sum_{h=1}^{K}\sum_{i=1}^{|S_t|}\sum_{j=1}^{|S_h|}\alpha_{ti}\bar{\boldsymbol{z}}(\boldsymbol{x}_{ti}, t)^T\bar{\boldsymbol{z}}(\boldsymbol{x}_{hj}, h)\alpha_{hj}, \\ \text{s.t.} \quad & \sum_{t=1}^{K}\sum_{i=1}^{|S_t|}\alpha_{ti}\boldsymbol{e}_t = \boldsymbol{e}, \\ & 0 \leq \alpha_{ti} \leq C_t, \quad i = 1, \ldots, |S_t|, \quad t = 1, \ldots, K. \end{aligned} \tag{30}$$

where $\alpha_{ti} \geq 0$ and $\alpha_{hj} \geq 0$ are Lagrange multipliers. The proof for Theorem 4 can refer to Sect. 8.4. Moreover, it has

$$\boldsymbol{w} = \frac{1}{2}\sum_{t=1}^{K}\sum_{i=1}^{|S_t|}\alpha_{ti}\bar{\boldsymbol{z}}(\boldsymbol{x}_{ti}, t). \tag{31}$$

By resolving the dual form (30), we can obtain α_{ti} and α_{hj}, and thereafter the value of $\mathbf{w}$ can be calculated according to Eq. (31). Let $I_t \in R^{K+1} = (0, \ldots, 0, 1, 0, \ldots, 0)^T$ $(t = 0, \ldots, K)$ be a column vector with the $(t+1)$th elements being 1 and the other elements being 0. Based on the defined column vector I_t, it is easy to deduce that $\mathbf{w}_0 = \mathbf{w}^T I_0$ and $\mathbf{v}_t = \sqrt{\frac{K}{C_t}}\mathbf{w}^T I_t$.

In the second step, we fix the values of $\mathbf{w}$ and ρ and optimize the learning problem (23) over $\Delta\mathbf{z}(\mathbf{x}_{hj}, h)$. To do this, we have Theorem 5 as follow.

Theorem 5 *Supposing that $\boldsymbol{w}$ and ρ are fixed to be $\overline{\boldsymbol{w}}$ and $\overline{\rho}$, respectively, the solution of $\Delta\boldsymbol{z}(\boldsymbol{x}_{hj}, h)$ for optimizing problem (23) is*

$$\Delta z(\boldsymbol{x}_{hj}, h) = \delta_{hj}\frac{\overline{\boldsymbol{w}}_0 + \overline{\boldsymbol{v}}_h}{\parallel \overline{\boldsymbol{w}}_0 + \overline{\boldsymbol{v}}_h \parallel} = \delta_{hj}\frac{\overline{\boldsymbol{w}}^T(I_0 + \sqrt{\frac{K}{C_h}}I_h)}{\parallel \overline{\boldsymbol{w}}^T(I_0 + \sqrt{\frac{K}{C_h}}I_h) \parallel},\ j = 1, \ldots, |S_h|. \tag{32}$$

The above two steps repeat alternatively until the termination criterion is met. When the optimization procedure stops, we can obtain the classifier $f_K(\mathbf{x}) = (\mathbf{w}_0 + \mathbf{v}_K)^T\mathbf{x} - \rho_K$ for the target task, and the obtained classifier is used to predict the unknown examples.

Let $\mathbf{w}^\phi$ be the weight vectors in the feature space. We can derive the objective function for nonlinear problems from problem (29), as follows:

$$\begin{aligned}
\min\ & ||\mathbf{w}^\phi||^2 - \rho^T\mathbf{e} + \sum_{t=1}^{K} C_t \sum_{i=1}^{|S_t|}\xi_{ti} \\
\text{s.t.}\ & (\mathbf{w}^\phi)^T\phi(\bar{\mathbf{z}}(\mathbf{x}_{ti}, t)) \geq \rho^T\mathbf{e}_t - \xi_{ti},\ \ i = 1, \ldots, |S_t|,\ \ t = 1, \ldots, K \\
& \xi_{ti} \geq 0.
\end{aligned} \tag{33}$$

Considering that the noise vector $\Delta\mathbf{x}_{ti}$ is fixed and $\bar{\mathbf{z}}(\mathbf{x}_{ti}, t)$ is known, problem (33) is a standard one-class SVM, which satisfies the representer theorem [46]. By applying the representer theorem, the weight vector $\mathbf{w}^\phi$ can be expressed as $\mathbf{w}^\phi = \sum_{j=1}^{n}\alpha_j K(\bar{\mathbf{z}}(\mathbf{x}_{hj}, h), \cdot)$ and it has $\mathbf{w}^\phi \cdot \phi(\mathbf{z}(\mathbf{x}_{ti}, t)) = \sum_{j=1}^{n}\alpha_j K(\bar{\mathbf{z}}(\mathbf{x}_{hj}, h), \mathbf{z}(\mathbf{x}_{ti}, t))$. The dual from can be obtained by substituting $\mathbf{w}^\phi$ into problem (33). After solving the dual form, the values of α_{ti}, $\mathbf{w}^\phi$ and ρ can be obtained.

Then, we fix $\mathbf{w}^\phi$ and ρ, and optimize $\sum_{t=1}^{K} C_t \sum_{i=1}^{|S_t|}\xi_{ti}$ over $\Delta\mathbf{x}_{ti}$. Since $\mathbf{w}^\phi$ and ρ are known, problem (33) is transformed into problem (34) after substituting $\mathbf{w}^\phi \cdot \phi(\mathbf{z}(\mathbf{x}_{ti}, t) + \Delta\mathbf{z}(\mathbf{x}_{ti}, t)) = \sum_{j=1}^{n}\alpha_j K(\bar{\mathbf{z}}(\mathbf{x}_{hj}, h), \mathbf{z}(\mathbf{x}_{ti}, t) + \Delta\mathbf{z}(\mathbf{x}_{ti}, t)))$.

$$\begin{aligned}
\min\ & \sum_{t=1}^{K} C_t \sum_{i=1}^{|S_t|}\xi_{ti} \\
\text{s.t.}\ & \sum_{j=1}^{n}\alpha_j K\left(\bar{\mathbf{z}}(\mathbf{x}_{hj}, h), \mathbf{z}(\mathbf{x}_{ti}, t) + \Delta\mathbf{z}(\mathbf{x}_{ti}, t)\right)) \geq \rho^T\mathbf{e}_t - \xi_{ti}, \\
& \xi_{ti} \geq 0,\ \ i = 1, \ldots, |S_t|,\ \ t = 1, \ldots, K.
\end{aligned} \tag{34}$$

Theorem 6 *In problem* (34)*, the optimal* $\Delta z(\boldsymbol{x}_{hj}, h)$ *is computed by*

$$\Delta z(\boldsymbol{x}_{hj}, h) = \delta_{hj} \frac{\widetilde{\boldsymbol{u}}_{hj}}{||\widetilde{\boldsymbol{u}}_{hj}||}, \quad j = 1, \ldots, |S_h|, \quad h = 1, \ldots, K \tag{35}$$

where it has

$$\widetilde{\boldsymbol{u}}_{hj} = \sum_{t=1}^{K} \sum_{i=1}^{|S_t|} \alpha_{ti} K'(z(\boldsymbol{x}_{ti}, t) + \Delta\bar{z}(\boldsymbol{x}_{ti}, t), z(\boldsymbol{x}_{hj}, h)).$$

In Theorem 6, $K'(\mathbf{z}(\mathbf{x}_{ti}, t) + \Delta\bar{\mathbf{z}}(\mathbf{x}_{ti}, t), \mathbf{z}(\mathbf{x}_{hj}, h))$ is the gradient of $K(\mathbf{z}(\mathbf{x}_{ti}, t) + \Delta\bar{\mathbf{z}}(\mathbf{x}_{ti}, t), \mathbf{z}(\mathbf{x}, \cdot))$ with respect to $\mathbf{z}(\mathbf{x}, \cdot)$ at points $\mathbf{z}(\mathbf{x}_{hj}, h)$. The proof for Theorem 6 can refer to Sect. 8.5. In the feature space, the nonlinear classifier for the target task is $f^K(\phi(\mathbf{x})) = (\mathbf{w}_0^\phi + \mathbf{v}_K^\phi)^T \phi(\mathbf{x}) - \rho_K$. By replacing $\bar{\mathbf{z}}(\mathbf{x}_{ti}, t)$ with $\phi(\bar{\mathbf{z}}(\mathbf{x}_{ti}, t))$, we can get the norm vector $\mathbf{w}^\phi$ in the feature space. Since it has $\mathbf{w}_0^\phi = (\mathbf{w}^\phi)^T I_0$ and $\mathbf{v}_K^\phi = (\mathbf{w}^\phi)^T I_K$, by substituting $\mathbf{w}_0^\phi$ and $\mathbf{v}_K^\phi$ into $f^K(\phi(\mathbf{x}))$, the nonlinear classifier $f^K(\phi(\mathbf{x}))$ for the target task can be computed as

$$f^K(\phi(\mathbf{x})) = \frac{1}{2}\left(I_0 + \sqrt{\frac{K}{C_K}} I_K\right)^T \sum_{t=1}^{K} \sum_{i=1}^{|S_t|} K(\bar{\mathbf{z}}(\mathbf{x}_{ti}, t), \mathbf{x}) - \rho_K \tag{36}$$

Our approach satisfies the representer theorem [46]. Similar to the linear cases, our approach for nonlinear classification contains two alternative steps. The first step is to fix the noise vector $\Delta\mathbf{x}_{ti}$ [i.e., $\Delta\mathbf{z}(\mathbf{x}_{ti}, t)$] and solve a QP problem (33) which is a standard one-class SVM and meets the representer theorem [46]. The second step is to fix $\mathbf{w}^\phi$ and ρ, and update the noise vector $\Delta\mathbf{x}_{ti}$ (i.e., $\Delta\mathbf{z}(\mathbf{x}_{ti}, t)$), as shown in problem (34). This step is computed based on $\mathbf{w}^\phi \cdot \phi(\cdot) = \sum_{j=1}^{n} \alpha_j K(\bar{\mathbf{z}}(\mathbf{x}_{hj}, h), \cdot)$, which has been obtained in the first step. Hence, thought our approach is non-convex, it satisfies the representer theorem.

6 Experiments

To investigate the effectiveness of our proposed UOCT-SVM approach, we conduct experiments on several real-world datasets. All experiments run on a laptop with 2.8 GHz processor and 3GB DRAM. The SVM-based algorithms are implemented based on LibSVM [47]. The objectives of our experiments are: (1) to evaluate the effectiveness of UOCT-SVM on transferring knowledge to the target task from one or more than one source tasks; (2) to investigate the sensitivity of UOCT-SVM to different percentages of data noise.

6.1 Baselines and metrics

We compare UOCT-SVM with the following baselines:

1. The first baseline is standard one-class SVM (OC-SVM) [1], which uses a hyperplane to separate the target class and the origin of the space. It is used to show the improvement of our approach over OC-SVM.
2. The second baseline is transfer learning-based one-class SVM (TLOC-SVM), which is a variant of our approach by excluding the uncertain data processing scheme. We set $\Delta\mathbf{x}_{ij} = 0$ and straightforwardly utilize problems (8) and (23) to train a transfer learning classifier with a single source task and multiple source tasks, respectively, without updating $\Delta\mathbf{x}_{ij}$.

This baseline is utilized to evaluate the ability of our approach on dealing with data uncertainty.

3. The third baseline is uncertain one-class SVM (UOC-SVM) [9], which builds one-class classifier to handle uncertain data. This baseline is used to investigate the capability of our approach on transferring knowledge from the source task to benefit the construction of classifiers on the target task.

The performance of classification systems is evaluated in terms of F-measure value [48]. We use it as the evaluation metric in the experiments. The F-measure value trades off the precision p and recall r, and it has $F = 2pr/(p + r)$. From this definition, we know that only when both the precision p and recall r are large, the F-measure value will exhibit a large value.

6.2 Dataset description and experimental setting

6.2.1 One-class classification datasets

To evaluate the effectiveness of our approach, we conduct experiments on three real-world datasets—Reuters-21578,[2] 20 Newsgroup,[3] and mushroom[4] datasets. These datasets are popularly used in the previous transfer learning work [21,28–30]. To fulfill the transfer learning purpose, we split and reorganize each dataset to generate the source task, which has the similar but different distribution to the test data, and the target task that has the same distribution with the test data.

The 20 Newsgroup and Reuters-21578 datasets have hierarchical structures. Taking the 20 Newsgroup dataset as an example, it has 7 top categories. Under the top categories, there are 20 sub-categories and each sub-category has 1,000 examples. Following the same routine in previous work [49], we generate the one-class transfer learning datasets based on the top categories. Specifically, we consider one sub-category as the target class in turn, and select a number of examples from the other top categories as the non-target class. To do this, a sub-category (a_1) from a top category (A) is selected and considered as the target class. The examples from the other top categories, i.e., those except for category (A), are treated as the nontarget class. Based on this, we generate the target class and the nontarget class for the target task. For the source task, we choose a sub-category (a_2) from the same top category (A), and consider this sub-category (a_2) as the target class. The sub-datasets generated from the 20 Newsgroup dataset are named as "NG.*", as shown in Table 1. In this table, "NG.os" indicates that in the "NG.os" sub-dataset, the sub-category "os" is considered as the target class for the target task, while the other sub-categories in the same top category are regarded as the target classes for the source tasks.

For the Reuters-21578 dataset, each top category has many sub-categories. For example, the top category "people" has 267 sub-categories and the size of each sub-category is not always large. As in Pan and Qiang [28] and Dai et al. [21,29], we reorganize the sub-categories within each top category. For a top category (A), all of the subcategories are reorganized into two parts (denoted as $a(1)$ and $a(2)$), and each part is approximately equal in sizes. Then, the reorganized sub-categories $a(1)$ and $a(2)$ are regarded as the target classes for the target task and the source task, respectively. Similar to the 20 Newsgroup dataset, the examples from the other top categories, i.e., those except for the top category (A), are treated as the nontarget class for the target task. The generated sub-datasets are named as "RT.*" in Table 1.

[2] Available at http://www.daviddlewis.com/resources/testcollections/.

[3] Available at http://people.csail.mit.edu/jrennie/20Newsgroups/.

[4] Available at http://archive.ics.uci.edu/ml/datasets/Mushroom.

Table 1 The categories contained in the target and source tasks for each sub-dataset

Dataset	Target task	Source task
NG.os	comp.os	comp.{graphics, ibm, mac }
NG.ibm	comp.ibm	comp.{graphics, os, mac }
NG.mac	comp.mac	comp.{graphics, os, ibm }
NG.graphics	comp.graphics	comp.{os, ibm, mac }
NG.autos	rec.autos	rec.{sport.baseball, sport.hokey }
NG.baseball	rec.sport.baseball	rec.{autos, sport.hokey }
NG.hokey	rec.sport.hokey	rec.{autos, sport.baseball }
NG.crypt	sci.crypt	sci.{med, space }
NG.med	sci.med	sci.{crypt, space }
NG.space	sci.space	sci.{crypt, med }
NG.religion	talk.religion	talk.politics.{guns, mideast }
NG.guns	talk.politics.guns	talk.{politics.mideast, religion }
NG.mideast	talk.politics.mideast	talk.{politics.guns, religion }
RT.orgs(1)	orgs(1).{ ... }	orgs(2).{ ... }
RT.orgs(2)	orgs(2).{ ... }	orgs(1).{ ... }
RT.people(1)	people(1).{ ... }	people(2).{ ... }
RT.people(2)	people(2).{ ... }	people(1).{ ... }
RT.place(1)	place(1).{ ... }	place(2).{ ... }
RT.place(2)	place(2).{ ... }	place(1).{ ... }
MR.edible(1)	edible(enlarging)	edible(tapering)
MR.edible(2)	edible(tapering)	edible(enlarging)
MR.poisonous(1)	poisonous(enlarging)	poisonous(tapering)
MR.poisonous(2)	poisonous(tapering)	poisonous(enlarging)

In the above operations, we generate the target classes, i.e., $a(1)$ and $a(2)$, from the same top category (A) for the target task and the source task, respectively, which guarantees that the two tasks are related. Otherwise, transfer learning may not improve, or may even hurt, the performance of the target task, which can be referred to as *negative transfer* [50].

For the mushroom dataset, since it does not have hierarchy, we follow the same routine in [29] to split the dataset based on the feature "stalk-shape". The mushroom dataset has two categories: "edible" and "poisonous", and the feature "stalk-shape" have two optional values: "enlarging" and "tapering". As in Dai et al. [29], we generate four sub-datasets, as shown in Table 1. For the MR.edible(1) sub-dataset, "edible(enlarging)" in the target task column and "edible(tapering)" in the source task column represent that for all the examples in the "edible" category, those whose values in the "stalk-shape" feature are equal to "enlarging" are considered as the target task, and those equivalent to "tapering" are regarded as the source task. The other sub-datasets have similar meanings.

6.2.2 Uncertain information generation

The above datasets are deterministic, and we need to model and involve uncertainty to these datasets. We follow the same operations in [51] to generate uncertain data, as follows.

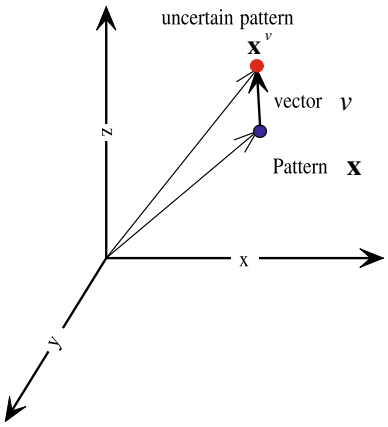


Fig. 2 Illustration of adding noises to the data example $\mathbf{x}$. $\mathbf{x}$ is the original example. $\mathbf{v}$ is the added noises. $\mathbf{x}^{\mathbf{v}}$ is the new example with added noises. By adding the noise vector $\mathbf{v}$, the new example $\mathbf{x}^{\mathbf{v}}$ has a certain deviation from the original example $\mathbf{x}$

To include uncertain data in the training set, we first compute the standard deviation σ_i^0 of the entire data along the ith dimension. In order to model the difference in noises on different dimensions, we define the standard deviation σ_i along the ith dimension which is randomly selected from the range $[0, 2\sigma_i^0]$. Then, for the ith dimension, we add noises from a random distribution with standard deviation σ_i. By doing this, an example $\mathbf{x}_{tj}$ is added with noises, which can be represented as a vector $\sigma^{\mathbf{x}_{tj}} = [\sigma_1^{\mathbf{x}_{tj}}, \sigma_2^{\mathbf{x}_{tj}}, \ldots, \sigma_{d-1}^{\mathbf{x}_{tj}}, \sigma_d^{\mathbf{x}_{tj}}]$. Here, d denotes the number of dimensions for a data example $\mathbf{x}_{tj}$, and $\sigma_i^{\mathbf{x}_{tj}}, i = 1, \ldots d$ represents the noises added into the ith dimension of the data example. Figure 2 illustrates the basic idea of this method. In this figure, $\mathbf{x}$ is the original example. $\mathbf{v}$ is the added noise. $\mathbf{x}^{\mathbf{v}}$ is the new example. By adding the noise $\mathbf{v}$, the new example $\mathbf{x}^{\mathbf{v}}$ has some deviations from the original example $\mathbf{x}$.

In the experiments, the RBF kernel $K(\mathbf{x}_{ti}, \mathbf{x}_{hj}) = \exp(-\| \mathbf{x}_{ti} - \mathbf{x}_{hj} \|_2^2 / 2\tau^2))$ is used. The parameter τ in the RBF kernel function is selected from 2^{-10} to 2^{10}. In our approach, $C_1, C_2, \ldots, C_{K-1}$ are regularized parameters associated with the $K-1$ source tasks, and C_K is with the target task. By adjusting the values of these parameters, we can make a tradeoff between the source tasks and the target task. If the regularized parameter of the target task C_K is larger than those of the source tasks C_i $(i = 1, \ldots, K-1)$, it prefers the target task to the source tasks. Otherwise, it prefers the source tasks to the target task. In transfer learning setting, the target task attracts more attentions than the source tasks, and we let $C_K > C_i (i = 1, \ldots, K-1)$. Moreover, for simplicity, we set $C_1 = \cdots = C_{K-1}$ and let it selected from 0 to 1,000. Likewise, C_K is picked up from 0 to 1,000. For the bound score δ_{ti} of example $\mathbf{x}_{ti}$, we compute it from the k-nearest neighbors of $\mathbf{x}_{ti}$ and set k equal to ten percentages of the training target examples. For parameter ε, we set it to be 0.1.

6.2.3 Performance comparison

For each sub-dataset in Table 1, we form the training set by randomly selecting ten percentages of the target examples from the target task and all the examples from the source tasks. This is because transfer learning usually assumes that there are insufficient training examples from the target task to learn the classifier. The remaining target examples and nontarget examples from the target task are used as the testing set. To avoid sampling bias, we repeat the above process ten times, and report the average F-measure values on the testing sets, as shown in Tables 2 and 3. Here, the noise percentage is set to be 40 %, i.e., 40 % examples being selected to add the noise.

Table 2 *F*-measure values for transfer learning problems with a single source task

Dataset	UOCT-SVM	UOC-SVM	TLOC-SVM	OC-SVM
MR.edible(1)	**87.51 ± 2.87**	84.36 ± 3.28	84.09 ± 3.48	81.97 ± 3.65
MR.edible(2)	**85.68 ± 2.14**	81.39 ± 2.35	82.55 ± 2.16	78.11 ± 2.79
MR.poisonous(1)	**82.85 ± 4.24**	80.14 ± 4.52	80.62 ± 3.82	78.19 ± 4.64
MR.poisonous(2)	**85.49 ± 3.32**	80.76 ± 3.16	81.84 ± 3.65	78.22 ± 3.86
RT.orgs(1)	**79.63 ± 4.56**	76.26 ± 4.95	74.01 ± 4.63	70.35 ± 5.17
RT.orgs(2)	**84.39 ± 2.54**	78.66 ± 2.73	80.83 ± 2.77	73.92 ± 3.08
RT.people(1)	**82.84 ± 4.23**	80.53 ± 3.58	76.98 ± 4.45	74.17 ± 4.75
RT.people(2)	**75.72 ± 3.46**	71.05 ± 3.29	72.15 ± 3.52	68.08 ± 4.29
RT.place(1)	**70.74 ± 3.82**	63.51 ± 4.39	66.78 ± 4.46	61.88 ± 4.67
RT.place(2)	**81.36 ± 2.79**	76.12 ± 2.86	75.82 ± 3.24	71.73 ± 3.58

The highest *F*-measure values are in bold

Table 3 *F*-measure values for transfer learning problems with multiple source tasks

Dataset	UOCT-SVM	UOC-SVM	TLOC-SVM	OC-SVM
NG.os	**76.63 ± 2.86**	72.16 ± 3.27	70.95 ± 3.34	67.57 ± 3.69
NG.ibm	**80.26 ± 3.44**	77.81 ± 3.19	76.75 ± 3.51	74.49 ± 3.95
NG.mac	**79.58 ± 3.18**	75.77 ± 3.58	75.43 ± 3.23	72.21 ± 3.86
NG.graphics	**83.57 ± 3.87**	78.26 ± 4.24	77.31 ± 4.54	73.31 ± 4.65
NG.hokey	**84.81 ± 3.24**	79.58 ± 3.59	81.19 ± 3.35	76.65 ± 3.72
NG.baseball	**78.79 ± 4.34**	76.05 ± 4.71	75.93 ± 4.62	71.97 ± 4.46
NG.autos	71.34 ± 2.91	**72.96 ± 2.63**	68.32 ± 3.15	69.35 ± 3.56
NG.crypt	**80.76 ± 3.18**	75.89 ± 3.35	78.18 ± 3.42	73.86 ± 4.03
NG.med	**78.74 ± 4.31**	73.59 ± 4.48	76.28 ± 4.38	69.88 ± 4.84
NG.space	**81.37 ± 2.39**	76.78 ± 2.92	76.24 ± 2.69	71.76 ± 3.34
NG.religion	**79.06 ± 3.76**	74.72 ± 4.39	75.33 ± 4.05	72.23 ± 4.58
NG.guns	75.25 ± 3.81	**77.41 ± 3.35**	71.88 ± 4.28	73.79 ± 4.51
NG.mideast	**81.31 ± 2.13**	78.94 ± 2.62	77.97 ± 2.39	75.92 ± 2.94

The highest *F*-measure values are in bold

Tables 2 and 3 show the average *F*-measure values for transfer learning problems with a single source task and multiple source tasks, respectively. It is observed that our proposed approach UOCT-SVM delivers explicitly better classification accuracy than UOC-SVM. Although both of UOCT-SVM and UOC-SVM can deal with data uncertainty, UOCT-SVM is capable of transferring knowledge from the source task to the target task such that a more accurate classifier can be built for the target task even when insufficient training examples from the target task are available. Moreover, UOCT-SVM outperforms TLOC-SVM and OC-SVM. UOCT-SVM is able to reduce the effect of noises on the decision boundary. As a result, the classifier learnt by UOCT-SVM can be more robust to noises and obtains better classification accuracy than TLOC-SVM and OC-SVM. In addition, it is observed from Table 3 that the classification accuracy of UOCT-SVM is lower than UOC-SVM on the NG.autos and NG.guns sub-datasets. This may be because the source tasks are not explicitly related to the target task. As pointed out by [50], transfer learning does not always improve the accuracy;

when the source tasks are irrelevant to the target task, it may lower the performance, which is called negative transfer.

6.2.4 Performance on different noise levels

We investigate the sensitivity of UOCT-SVM, UOC-SVM, TLOC-SVM and OC-SVM to data noise. Figure 3 illustrates the variation of F-measure values when the percentage of noises increases from 20 to 100 % on part of the sub-datasets. The x-axis stands for the percentage of noises added to the training data. The y-axis represents the average F-measure values. It is seen that the F-measure values decrease with the increasing of noise percentages. This may be due to the fact that when the percentage of noise increases, the target class potentially becomes less distinguishable from the nontarget class. However, it is clearly to see that UOCT-SVM can deliver consistently higher F-measure values than OC-SVM and UOC-SVM with different percentages of noises. Compared to UOC-SVM, UOCT-SVM has the lower decrease of F-measure values by transferring knowledge from the source tasks to the target task. In contrast with TLOC-SVM and OC-SVM, UOCT-SVM still attains markedly better F-measure values when the percentage of noises increases from 20 to 100 %, which implies that UOCT-SVM is effective to reduce the effect of noises. The other sub-datasets have similar observations.

6.2.5 Running time analysis

So far, we have investigated the classification accuracy of UOCT-SVM and the baselines. It is interesting to compare the running time. Figure 4 presents the training time of OC-SVM, UOC-SVM, TLOC-SVM and UOCT-SVM on the experimental sub-datasets. It is observed that OC-SVM is the most efficient method. OC-SVM does not deal with data uncertainty and transfer learning scenarios when training the classifier. As a result, it trains faster, but has lower classification accuracy than UOCT-SVM. TLOC-SVM is the second efficient method of which the running time is slightly lower than OC-SVM. UOC-SVM is the third efficient method and UOCT-SVM is slower than UOC-SVM. UOCT-SVM redefines the training examples as $\mathbf{z}(\mathbf{x}_{ti}, t)$, which has a larger number of dimensions than the original example $\mathbf{x}_{ti}$ and takes up more computational time. However, the F-measure value of UOCT-SVM is explicitly higher than UOC-SVM on most of the experimental sub-datasets, as shown in Tables 2 and 3. For example, on the RT.orgs(2) sub-dataset, the F-measure value of UOCT-SVM is 84.39 %, which is higher than UOC-SVM (78.66 %) at 5.73 %.

6.2.6 Sensitivity to regularization parameters

In problem (7), C_1 and C_2 are regularization parameters associated with the source task and the target task, respectively. They control the discrepancies between the global optimal boundary $\mathbf{w}$ and the local optimal boundaries $\mathbf{w}+\mathbf{v}_1$ (for the source task) and $\mathbf{w}+\mathbf{v}_2$ (for the target task). When a relative lager value of C_1 than C_2 is set, the global optimal solution $\mathbf{w}$ biases toward the source task, and vice versa. If we let $C_1 \gg C_2$, e.g., $\frac{C_1}{C_2} > 1{,}000$, the value of $\mathbf{v}_1$ approaches to zero and problem (7) degrades to a standard SVM. The global optimal boundary approximates the classification boundary of the source task. If we let $C_1 \ll C_2$, the global optimal boundary approaches to the boundary of the target task.

Furthermore, we take the MR.edible(1) sub-dataset as an example and investigate the performance variation in our approach with different values of C_1 and C_2. In Fig. 5a, we fix

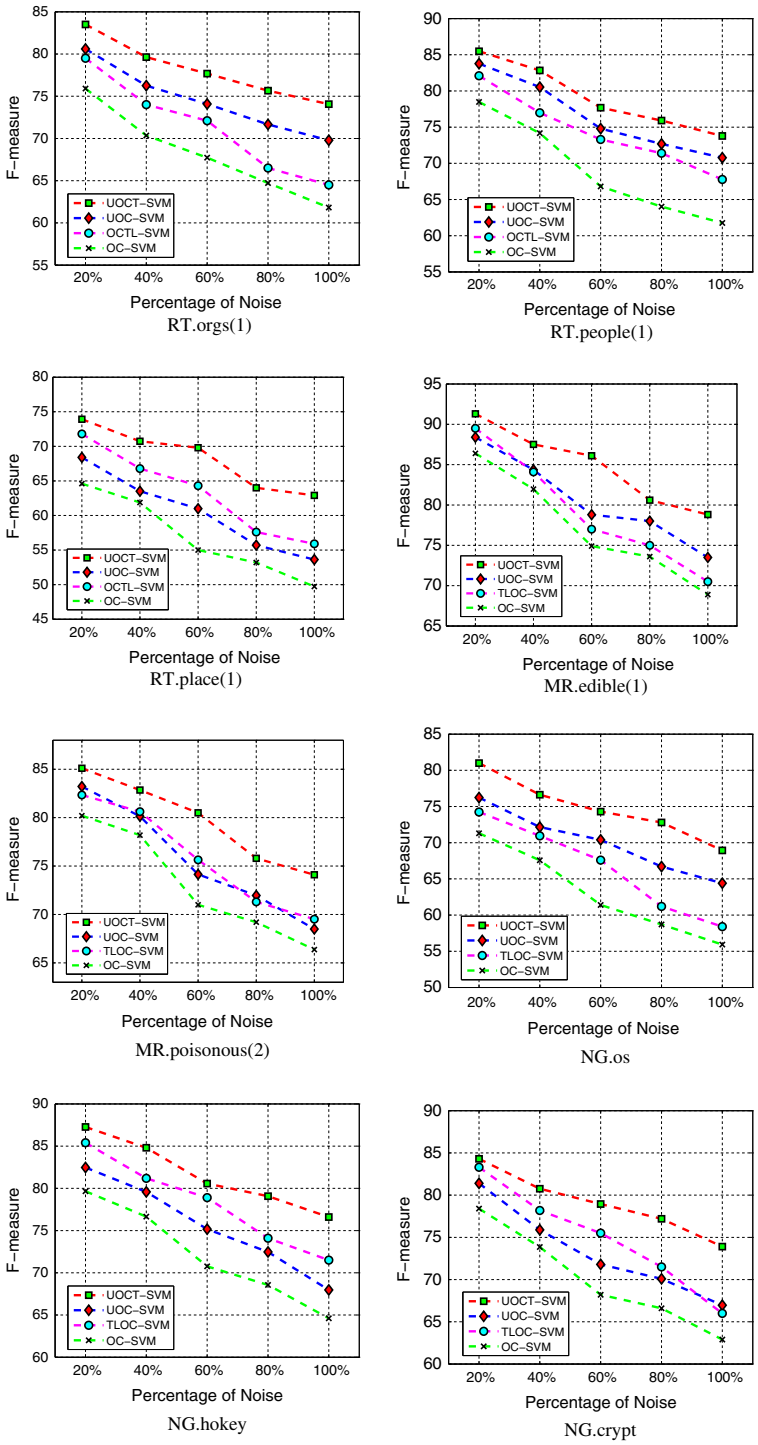


Fig. 3 Performance of OC-SVM, UOC-SVM and UOCT-SVM at different noise levels

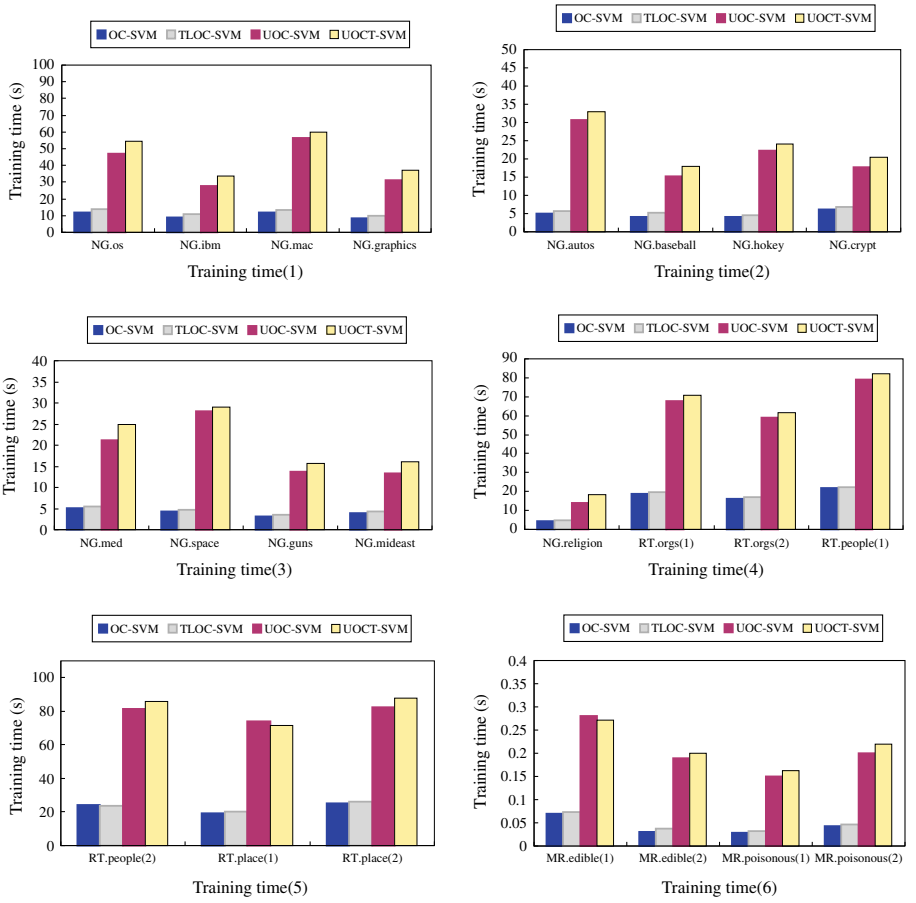
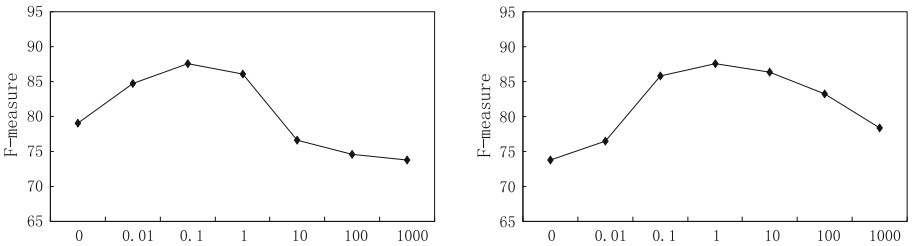


Fig. 4 Training time of OC-SVM, UOC-SVM and UOCT-SVM

$C_2 = 1$ and report the F-measure values when C_1 increases from 0 to 1,000. It is observed that the best F-measure value is achieved when C_1 is equal to 0.1. As the increase of C_1, there is a relative obvious decrease when C_1 is larger than 1 (namely C_2). In Fig. 5b, we set $C_1 = 0.1$ and present the results with varying values of C_2. Similarly, when C_2 is equivalent to 1, the highest F-measure value is obtained. When C_2 is less than 0.1 (namely C_1), a noticeable decline is observed. This is because, when C_2 is smaller than C_1, the target task is considered to be less important and the global optimal solution bias toward the source task. Therefore, in order to let the global optimal solution bias toward the target task, a larger value of C_2 than C_1 is usually set. In the experiments, we empirically find that a relative satisfactory result can be obtained when the value of $\frac{C_2}{C_1}$ is around 10. In real-world applications, we need validation data to find the optimal C_1 and C_2 for different datasets.

6.2.7 Performance comparison on real-world uncertain datasets

In the above sections, we artificially add the noises to the experimental datasets and evaluate the sensitivity of our approach to different levels of noises. In the following, we will test our



(a) Results with varying C_1 (C_2=1) **(b)** Results with varying C_2 (C_1=0.1)

Fig. 5 Performance on different values of the regularization parameters C_1 and C_2

method on two real-world uncertain datasets: Isolet dataset[5] for speech classification and Localization Data for Person Activity (LDPA) dataset[6] for sensor-based abnormal activity prediction.

The first one is the Isolet dataset, which is a speech utterance classification dataset and naturally contains noisy information since the utterance data is collected from 150 different speakers and they vary greatly in the way of utterances. The utterance for the same content may be different. The speakers utter the characters in the English alphabet and the Isolet dataset contains 7,797 examples in total. Each example contains 617 features extracted from the utterance data, and the exact feature description can be found in [52]. The task is to classify which English alphabet is uttered.

In the experiments, we first select out the examples related to English alphabets "S", "X", "M" and "N", respectively, and form four sub-datasets S_S, S_X, S_M and S_N. Second, S_S, S_X, S_M and S_N are divided into two groups: $\{S_S, S_X\}$ and $\{S_M, S_N\}$. Third, we form four one-class transfer learning problems, i.e., Isolet(S), Isolet(X), Isolet(M) and Isolet(N), by treating one sub-dataset as the target task and the other sub-dataset as the source task in turn for each group. For the Isolet(S) sub-dataset, we choose ten percentages of the examples from S_S as the target task and all the examples from S_X as the source task to form the training set. In the testing set, the remaining examples from S_S are considered as the target class, and three hundred examples randomly selected from the Isolet dataset, expect for those related to "S" and "X", are treated as the nontarget class. This process is repeated ten times, and the average F-measure values are reported. The task of Isolet(S) is to predict whether a test utterance is the English alphabet "S". Isolet(S) is a one-class transfer learning problem by considers S_X as the source task and S_S as the target task. On one hand, S_X and S_S are different because they are related to two distinctive English alphabets "X" and S, respectively. On the other hand, S_X and S_S are related to each other since the utterances of English alphabets "X" and S are similar to some extent. Hence, we can build the one-class classifier on S_S by transferring the knowledge from S_X. Likewise, Isolet(X) treats S_X as the target task and S_S as the source task. Isolet(M) considers S_M as the target task and S_N as the source task. Isolet(N) treats S_N as the target task and S_M as the source task.

The second one is the Localization Data for Person Activity (LDPA) dataset, which is a sensor-based dataset, aiming at determining the abnormal activities from the normal activities. It contains recordings of five people performing eleven activities: "walking", "falling", "laundry", "lying down", "on all fours", "standing up from sitting on the ground", etc. Following the same operations in [53], the eleven activities are classified into two classes:

[5] Available from http://archive.ics.uci.edu/ml/datasets/ISOLET.

[6] Available from http://dis.ijs.si/confidence/dataset.html.

Table 4 F-measure values on the Isolet and LDPA sub-datasets

Dataset	UOCT-SVM	UOC-SVM	TLOC-SVM	OC-SVM
Isolet(S)	**90.06 ± 2.14**	85.94 ± 2.51	87.29 ± 2.26	83.58 ± 2.47
Isolet(X)	**88.82 ± 3.36**	87.27 ± 3.49	85.75 ± 3.34	83.64 ± 3.97
Isolet(M)	**93.26 ± 1.86**	91.18 ± 1.97	89.91 ± 2.25	89.01 ± 2.62
Isolet(N)	**91.32 ± 2.38**	89.29 ± 2.56	88.69 ± 2.72	85.99 ± 3.18
LDPA(1)	**86.98 ± 3.14**	84.12 ± 3.42	83.66 ± 3.53	80.42 ± 3.76
LDPA(2)	**77.84 ± 3.61**	74.75 ± 3.58	73.86 ± 3.65	71.53 ± 3.95
LDPA(3)	**83.11 ± 2.54**	78.28 ± 2.58	79.87 ± 2.87	74.55 ± 3.07
LDPA(4)	**84.46 ± 2.43**	79.48 ± 2.77	81.82 ± 2.51	78.64 ± 3.14
LDPA(5)	**79.46 ± 3.17**	75.72 ± 3.23	78.25 ± 3.28	73.27 ± 3.76

The highest F-measure values are in bold

activities "falling", "on all fours", "sitting on the ground" and "standing up from sitting on the ground" belonging to the nontarget class, and the other activities belonging to the target class. Each person wears four localization sensors (ankle left, ankle right, belt and chest) and performs different activities. The localization sensors record the x, y and z coordinates. The LDPA dataset is relatively large, containing 164,860 examples, and we utilize around one tenth of the dataset, i.e., 16,486 examples. Each example contains eight features, including x coordinate of sensors, y coordinate of sensors, z coordinate of sensors, etc.

We put the normal data and abnormal data related to the ith person into subset S_i^+ and S_i^- ($i = 1, 2, \ldots, 5$), respectively. Considering that the five people may not perform the same activity at exactly the same locations, the collected localization data can differ even for the same activity. Moreover, the five people may in distinctive heights and body shapes, which brings further difference to the localization data. Hence, we consider the data related to one person as a learning task and obtain five tasks: one person for one task. Then, we treat each task as the target task in turn, and acquire five one-class transfer learning problems, denoted as "LDPA(1)", "LDPA(2)", "LDPA(3)", "LDPA(4)" and "LDPA(5)". In "LDPA(1)", S_1^+ is considered as the target task, and the source task is randomly selected from the remaining four subsets S_2^+, S_3^+, S_4^+ and S_5^+. Ten percentages of examples from S_i^+ and all examples from the source task are used to form the training set. In the testing set, the remaining examples from S_i^+ are treated as the target class, and S_i^- is as the non-target class. "LDPA(2)", "LDPA(3)", "LDPA(4)" and "LDPA(5)" are formed in a similar way to "LDPA(1)".

Table 4 presents the F-measure values of our approach UOCT-SVM and the baselines on the Isolet and LDPA sub-datasets. After investigating the details in Table 4, it is seen that UOCT-SVM obtains the best F-measure values on all the sub-datasets. Taking the Isolet(S) sub-dataset as an example, the F-measure value of UOCT-SVM is 90.06 %, which gains a minimum of 2.77 % and up to 6.48 % improvements, relative to UOC-SVM, TLOC-SVM and OC-SVM. TLOC-SVM and OC-SVM do not take the data uncertainty into account, and the better performance of UOCT-SVM over TLOC-SVM and OC-SVM indicates that the experimental real-world datasets may contain uncertain information, and UOCT-SVM is capable of dealing with data uncertainty to refine the decision boundary and improve the learning accuracy.

7 Conclusions and future work

Most of the existing one-class classification methods assume that there are sufficient training examples, and they do not contain any uncertain information. Nevertheless, these assumptions

cannot always be met in real-world applications. In this paper, we propose a novel approach, termed as uncertain one-class transfer learning with support vector machine (UOCT-SVM). UOCT-SVM explicitly deals with uncertain data by introducing a boundary score for each example. Then, the uncertain examples, together with their boundary scores, are incorporated into an one-class transfer learning model. An iterative framework is put forward to solve the optimization problem such that we can obtain an accurate classifier for the target task by transferring knowledge from the source task. Extensive experiments has demonstrated the effectiveness of our approach.

In the future, we plan to exploit more optimization methods to accelerate the training efficiency of UOCT-SVM. In the experiments, we utilize LibSVM [47] to solve the QP problems (9) and (24), and the time complexity of solving a QP problem with n examples is about $O(n^2)$. Assuming that our algorithm terminates after T iterations, the time complexity is approximately $O(Tn^2)$. When the dataset size is large, the time complexity may be high. As the future work, we would like to speed up our approach by employing more efficient optimization methods [54–56], such as Pegasos [57] and NORMA [58] where the running time of solving a QP problem is linearly with the number of nonzero features in each example, and does not depend directly on the dataset size. They can be utilized to solve the QP problems (9) and (24) on large-scale datasets.

Acknowledgments This work is supported by Natural Science Foundation of China (61070033, 61203280, 61202270), Guangdong Natural Science Funds for Distinguished Young Scholar (S2013050014133), Natural Science Foundation of Guangdong province (9251009001000005, S2012040007078), Specialized Research Fund for the Doctoral Program of Higher Education (20124420120004), Science and Technology Plan Project of Guangzhou City(12C42111607, 201200000031,2012J5100054), Science and Technology Plan Project of Panyu District Guangzhou (2012-Z-03-67), Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, GDUT Overseas Outstanding Doctoral Fund (405120095), 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.

8 Appendix

8.1 Proof for Theorem 1

Assume that $\alpha_{1i} \geq 0$, $\alpha_{2j} \geq 0$, $\beta_{1i} \geq 0$ and $\beta_{2j} \geq 0$ are Lagrange multipliers. The Lagrange function of problem (8) can be given as

$$\begin{aligned} L = &||\mathbf{w}_0||^2 + C_1||\mathbf{v}_1||^2 + C_2||\mathbf{v}_2||^2 - \rho_1 - \rho_2 + C_1\sum_{i=1}^{|S_1|}\xi_{1i} + C_2\sum_{j=1}^{|S_2|}\xi_{2j} \\ &+ \sum_{i=1}^{|S_1|}\alpha_{1i}[\rho_1 - \xi_{1i} - (\mathbf{w}_0 + \mathbf{v}_1)^T\bar{\mathbf{x}}_{1i}] + \sum_{j=1}^{|S_2|}\alpha_{2j}[\rho_2 - \xi_{2j} - (\mathbf{w}_0 + \mathbf{v}_2)^T\bar{\mathbf{x}}_{2j}] \\ &- \sum_{i=1}^{|S_1|}\beta_{1i}\xi_{1i} - \sum_{j=1}^{|S_2|}\beta_{2j}\xi_{2j} \end{aligned} \tag{37}$$

where it has $\bar{\mathbf{x}}_{1i} = \mathbf{x}_{1i} + \Delta\bar{\mathbf{x}}_{1i}$ and $\bar{\mathbf{x}}_{2j} = \mathbf{x}_{2j} + \Delta\bar{\mathbf{x}}_{2j}$.

By differentiating the Lagrange function (37) with $\mathbf{w}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$, ρ_1, ρ_2, ξ_{1i} and ξ_{2j}, respectively, the following equations can be obtained.

$$\frac{\partial L}{\partial \mathbf{w}_0} = 2\mathbf{w}_0 - \sum_{i=1}^{|S_1|} \alpha_{i1}\bar{\mathbf{x}}_{i1} - \sum_{j=1}^{|S_2|} \alpha_{2j}\bar{\mathbf{x}}_{2j} = 0, \tag{38}$$

$$\frac{\partial L}{\partial \mathbf{v}_1} = 2C_1\mathbf{v}_1 - \sum_{i=1}^{|S_1|} \alpha_{1i}\bar{\mathbf{x}}_{1i} = 0, \tag{39}$$

$$\frac{\partial L}{\partial \mathbf{v}_2} = 2C_2\mathbf{v}_2 - \sum_{j=1}^{|S_2|} \alpha_{2j}\bar{\mathbf{x}}_{2j} = 0, \tag{40}$$

$$\frac{\partial L}{\partial \rho_1} = -1 + \sum_{i=1}^{|S_1|} \alpha_{1i} = 0, \tag{41}$$

$$\frac{\partial L}{\partial \rho_2} = -1 + \sum_{j=1}^{|S_2|} \alpha_{2j} = 0, \tag{42}$$

$$\frac{\partial L}{\partial \xi_{1i}} = C_1 - \alpha_{1i} - \beta_{1i} = 0, \quad i = 1, \ldots, |S_1| \tag{43}$$

$$\frac{\partial L}{\partial \xi_{2j}} = C_2 - \alpha_{2j} - \beta_{2j} = 0, \quad j = 1, \ldots, |S_2| \tag{44}$$

From Eqs. (38)–(44), it is easy to deduce that

$$\mathbf{w}_0 = \frac{1}{2}\left(\sum_{i=1}^{|S_1|} \alpha_{1i}\bar{\mathbf{x}}_{1i} + \sum_{j=1}^{|S_2|} \alpha_{2j}\bar{\mathbf{x}}_{2j}\right), \tag{45}$$

$$\mathbf{v}_1 = \frac{1}{2C_1}\sum_{i=1}^{|S_1|} \alpha_{1i}\bar{\mathbf{x}}_{1i}, \tag{46}$$

$$\mathbf{v}_2 = \frac{1}{2C_2}\sum_{j=1}^{|S_2|} \alpha_{2j}\bar{\mathbf{x}}_{2j}, \tag{47}$$

$$\sum_{i=1}^{|S_1|} \alpha_{1i} = 1, \tag{48}$$

$$\sum_{j=1}^{|S_2|} \alpha_{2j} = 1, \tag{49}$$

$$C_1 = \alpha_{1i} + \beta_{1i}, \quad i = 1, \ldots, |S_1| \tag{50}$$

$$C_2 = \alpha_{2j} + \beta_{2j}, \quad j = 1, \ldots, |S_2| \tag{51}$$

Since it has $\beta_{1i} \geq 0$ and $\beta_{2j} \geq 0$, from (50) and (51), we can obtain

$$0 \leq \alpha_{1i} \leq C_1, \quad i = 1, \ldots, |S_1| \tag{52}$$

$$0 \leq \alpha_{2j} \leq C_2, \quad j = 1, \ldots, |S_2| \tag{53}$$

By substituting (38)–(53) into the Lagrange function (37), the dual form of problem (8) can be written as

$$\begin{aligned}
\max \quad & -\frac{1}{2}\sum_{i=1}^{|S_1|}\sum_{j=1}^{|S_2|}\alpha_{1i}\overline{\mathbf{x}}_{1i}^T\overline{\mathbf{x}}_{2j}\alpha_{2j} - \frac{C_1+1}{4C_1}\sum_{h=1}^{|S_1|}\sum_{g=1}^{|S_1|}\alpha_{1h}\overline{\mathbf{x}}_{1h}^T\overline{\mathbf{x}}_{1g}\alpha_{1g} \\
& -\frac{C_2+1}{4C_2}\sum_{p=1}^{|S_2|}\sum_{k=1}^{|S_2|}\alpha_{2p}\overline{\mathbf{x}}_{2p}^T\overline{\mathbf{x}}_{2k}\alpha_{2k} \\
\text{s.t.} \quad & \sum_{i=1}^{|S_1|}\alpha_{1i} = 1, \quad 0 \leq \alpha_{1i} \leq C_1, \quad i = 1, \ldots, |S_1| \\
& \sum_{j=1}^{|S_2|}\alpha_{2j} = 1, \quad 0 \leq \alpha_{2j} \leq C_2, \quad j = 1, \ldots, |S_2|
\end{aligned}$$

□

8.2 Proof for Theorem 2

In Theorem 2, we fix $\mathbf{w}_0, \mathbf{v}_1, \mathbf{v}_2, \rho_1$ and ρ_2 to be $\overline{\mathbf{w}}_0, \overline{\mathbf{v}}_1, \overline{\mathbf{v}}_2, \overline{\rho}_1$ and $\overline{\rho}_2$, respectively, and attempt to minimize the value of the objective function (7) by optimizing $\Delta\mathbf{x}_{1i}$ and $\Delta\mathbf{x}_{2j}$. From (7), the objective function's value is determined by $\sum_{t=1}^{2}\sum_{i=1}^{|S_t|}\xi_{ti}$ since $\mathbf{w}_0, \mathbf{v}_1, \mathbf{v}_2, \rho_1$ and ρ_2 are fixed. Hence, we need to optimize $\Delta\mathbf{x}_{1i}$ and $\Delta\mathbf{x}_{2j}$ to minimize $\sum_{t=1}^{2}\sum_{i=1}^{|S_t|}\xi_{ti}$.

Each training example $\mathbf{x}_{ti}$ $(i = 1, \ldots, |S_t|, t = 1, 2)$ is associated with an error term ξ_{ti} and the minimization of $\sum_{t=1}^{2}\sum_{i=1}^{|S_t|}\xi_{ti}$ can be decomposed into subproblems of minimizing each error term ξ_{ti}:

$$\begin{aligned}
\xi_{ti} &= \max\left\{0, \rho_t - (\overline{\mathbf{w}}_0 + \overline{\mathbf{v}}_t)^T(\mathbf{x}_{ti} + \Delta\mathbf{x}_{ti})\right\} \\
&= \max\left\{0, \rho_t - (\overline{\mathbf{w}}_0 + \overline{\mathbf{v}}_t)^T\mathbf{x}_{ti} - (\overline{\mathbf{w}}_0 + \overline{\mathbf{v}}_t)^T\Delta\mathbf{x}_{ti}\right\}
\end{aligned} \tag{54}$$

From Eq. (54), it is seen that we can minimize ξ_{ti} by maximizing $(\overline{\mathbf{w}}_0 + \overline{\mathbf{v}}_t)^T\Delta\mathbf{x}_{ti}$. According to the Cauchy–Schwarz inequality [59], it has

$$-||\overline{\mathbf{w}}_0 + \overline{\mathbf{v}}_t|| \cdot ||\mathbf{x}_{ti}|| \leq (\overline{\mathbf{w}}_0 + \overline{\mathbf{v}}_t)^T\Delta\mathbf{x}_{ti} \leq ||\overline{\mathbf{w}}_0 + \overline{\mathbf{v}}_t|| \cdot ||\mathbf{x}_{ti}|| \tag{55}$$

In Eq. (55) becomes equation if and only if $\Delta\mathbf{x}_{ti} = c(\overline{\mathbf{w}}_0 + \overline{\mathbf{v}}_t)$, where c is a constant number. Since $\Delta\mathbf{x}_{ti}$ is bounded by δ_{ti}, the optimal value of $\Delta\mathbf{x}_{ti}$ is

$$\Delta\mathbf{x}_{ti} = \delta_{ti}\frac{\overline{\mathbf{w}}_0 + \overline{\mathbf{v}}_t}{||\overline{\mathbf{w}}_0 + \overline{\mathbf{v}}_t||}, \quad i = 1, \ldots, |S_t|, \quad t = 1, 2. \tag{56}$$

□

8.3 Proof for Theorem 3

We fix $\overline{\mathbf{w}}_0, \overline{\mathbf{v}}_1, \overline{\mathbf{v}}_2, \overline{\rho}_1$ and $\overline{\rho}_2$, and focus on minimizing each $\xi = \max\{0, \overline{\rho}_t - \frac{1}{2}\sum_{h=1}^{2}\sum_{j=1}^{|S_h|}\alpha_{hj}K(\mathbf{x}_{hj} + \Delta\overline{\mathbf{x}}_{hj}, \mathbf{x} + \Delta\mathbf{x}) - \frac{1}{2C_t}\sum_{j=1}^{|S_t|}\alpha_{tj}K(\mathbf{x}_{tj} + \Delta\overline{\mathbf{x}}_{tj}, \mathbf{x} + \Delta\mathbf{x})$ $(\mathbf{x} \in S_t, t = 1, 2)$ over $\Delta\mathbf{x}$. According to the first order Taylor expansion of $K(\cdot)$ in Eq. (21), it is easy to deduce

$$\begin{aligned}
&\frac{1}{2}\sum_{h=1}^{2}\sum_{j=1}^{|S_h|}\alpha_{hj}K(\mathbf{x}_{hj}+\Delta\bar{\mathbf{x}}_{hj},\mathbf{x}+\Delta\mathbf{x})+\frac{1}{2C_t}\sum_{j=1}^{|S_t|}\alpha_{tj}K(\mathbf{x}_{tj}+\Delta\bar{\mathbf{x}}_{tj},\mathbf{x}+\Delta\mathbf{x})\\
&=\frac{1}{2}\sum_{h=1}^{2}\sum_{j=1}^{|S_h|}\alpha_{hj}K(\mathbf{x}_{hj}+\Delta\bar{\mathbf{x}}_{hj},\mathbf{x})+\Delta\mathbf{x}^T\frac{1}{2}\sum_{h=1}^{2}\sum_{j=1}^{|S_h|}\alpha_{hj}K'(\mathbf{x}_{hj}+\Delta\bar{\mathbf{x}}_{hj},\mathbf{x})\\
&\quad+\frac{1}{2C_t}\sum_{j=1}^{|S_t|}\alpha_{tj}K(\mathbf{x}_{tj}+\Delta\bar{\mathbf{x}}_{tj},\mathbf{x})+\Delta\mathbf{x}^T\frac{1}{2C_t}\sum_{j=1}^{|S_t|}\alpha_{tj}K'(\mathbf{x}_{tj}+\Delta\bar{\mathbf{x}}_{tj},\mathbf{x})\\
&=\frac{1}{2}\sum_{h=1}^{2}\sum_{j=1}^{|S_h|}\alpha_{hj}K(\mathbf{x}_{hj}+\Delta\bar{\mathbf{x}}_{hj},\mathbf{x})+\frac{1}{2C_t}\sum_{j=1}^{|S_t|}\alpha_{tj}K(\mathbf{x}_{tj}+\Delta\bar{\mathbf{x}}_{tj},\mathbf{x})\\
&\quad+\Delta\mathbf{x}^T\left[\frac{1}{2}\sum_{h=1}^{2}\sum_{j=1}^{|S_h|}\alpha_{hj}K'(\mathbf{x}_{hj}+\Delta\bar{\mathbf{x}}_{hj},\mathbf{x})+\frac{1}{2C_t}\sum_{j=1}^{|S_t|}\alpha_{tj}K'(\mathbf{x}_{tj}+\Delta\bar{\mathbf{x}}_{tj},\mathbf{x})\right] \qquad (57)
\end{aligned}$$

Similar to Sect. 8.2, by using the Cauchy–Schwarz inequality, the optimal value of $\Delta\mathbf{x}_{ti}$ is as follows

$$\Delta\mathbf{x}_{ti}=\delta_{ti}\frac{\mathbf{u}_{ti}}{||\mathbf{u}_{ti}||},\quad t=1,2$$

where it has

$$\mathbf{u}_{ti}=\frac{1}{2}\sum_{h=1}^{2}\sum_{j=1}^{|S_h|}\alpha_{hj}K'(\mathbf{x}_{hj}+\Delta\bar{\mathbf{x}}_{hj},\mathbf{x})+\frac{1}{2C_t}\sum_{j=1}^{|S_t|}\alpha_{tj}K'(\mathbf{x}_{tj}+\Delta\bar{\mathbf{x}}_{tj},\mathbf{x}).$$

□

8.4 Proof for Theorem 4

Let $\alpha_{ti}\geq 0$ and $\beta_{ti}\geq 0$ be Lagrange multipliers. Based on the Lagrange multipliers, the Lagrange function of problem (29) can be given as

$$\begin{aligned}
L=&||\mathbf{w}||^2-\rho^T\mathbf{e}+\sum_{t=1}^{K}C_t\sum_{j=1}^{|S_t|}\xi_{ti}\\
&+\sum_{t=1}^{K}\sum_{i=1}^{|S_t|}\alpha_{ti}(\rho^T\mathbf{e}_t-\xi_{ti}-\mathbf{w}^T\bar{\mathbf{z}}(\mathbf{x}_{ti},t))-\sum_{t=1}^{K}\sum_{i=1}^{|S_t|}\beta_{ti} \qquad (58)
\end{aligned}$$

Differentiating the Lagrange function (58) with $\mathbf{w}$, ρ, ξ_{ti} leads to

$$\frac{\partial L}{\partial\mathbf{w}}=2\mathbf{w}-\sum_{t=1}^{K}\sum_{i=1}^{|S_t|}\alpha_{ti}\bar{\mathbf{z}}(\mathbf{x}_{ti},t)=0 \qquad (59)$$

$$\frac{\partial L}{\partial\rho}=-\mathbf{e}+\sum_{t=1}^{K}\sum_{i=1}^{|S_t|}\alpha_{ti}\mathbf{e}_t=0, \qquad (60)$$

$$\frac{\partial L}{\partial\xi_{ti}}=C_t-\alpha_{ti}-\beta_{ti}=0. \qquad (61)$$

According to Eqs. (59)–(61), we can obtain

$$\mathbf{w} = \frac{1}{2}\sum_{t=1}^{K}\sum_{i=1}^{|S_t|}\alpha_{ti}\overline{\mathbf{z}}(\mathbf{x}_{ti}, t), \tag{62}$$

$$\sum_{t=1}^{K}\sum_{i=1}^{|S_t|}\alpha_{ti}\mathbf{e}_t = \mathbf{e}, \tag{63}$$

$$0 \le \alpha_{ti} \le C_t. \tag{64}$$

By substituting (62)–(64) to problem (29), the dual form can be given as

$$\begin{aligned} \max \ & -\frac{1}{4}\sum_{t=1}^{K}\sum_{h=1}^{K}\sum_{i=1}^{|S_t|}\sum_{j=1}^{|S_h|}\alpha_{ti}\overline{\mathbf{z}}(\mathbf{x}_{ti}, t)^T\overline{\mathbf{z}}(\mathbf{x}_{hj}, h)\alpha_{hj}, \\ \text{s.t.} \ & \sum_{t=1}^{K}\sum_{j=1}^{|S_t|}\alpha_{ti}\mathbf{e}_t = \mathbf{e}, \\ & 0 \le \alpha_{ti} \le C_t, \ \ i = 1, \ldots, |S_t|, \ \ t = 1, \ldots, K. \end{aligned} \tag{65}$$

□

8.5 Proof for Theorem 6

We fix $\mathbf{w}^{\phi}$ and ρ to be $\overline{\mathbf{w}}^{\phi}$ and $\overline{\rho}$, respectively, and minimize each $\xi_{hj} = \max\{0, \overline{\rho}_h^T\mathbf{e}_h - (\mathbf{w}^{\phi})^T\phi(\overline{\mathbf{z}}(\mathbf{x}_{hj}, h))\}$ ($\mathbf{x}_{hj} \in S_h, h = 1, \ldots, K$) over $\Delta\mathbf{x}_{hj}$. Since $\overline{\rho}_h^T\mathbf{e}_h$ is known, we minimize ξ_{hj} by maximizing $(\mathbf{w}^{\phi})^T\phi(\overline{\mathbf{z}}(\mathbf{x}_{hj}, h))$. Replacing $\overline{\mathbf{z}}(\mathbf{x}_{hj}, h)$ with $\phi(\overline{\mathbf{z}}(\mathbf{x}_{hj}, h))$ in Eq. (31) leads to

$$\mathbf{w}^{\phi} = \frac{1}{2}\sum_{t=1}^{K}\sum_{i=1}^{|S_t|}\alpha_{ti}\phi(\overline{\mathbf{z}}(\mathbf{x}_{ti}, t)) \tag{66}$$

By employing the first order Taylor expansion of $K(\cdot)$ in Eq. (21) and substituting Eq. (66) into $(\mathbf{w}^{\phi})^T\phi(\overline{\mathbf{z}}(\mathbf{x}_{hj}, h))$, it has

$$\begin{aligned} & (\mathbf{w}^{\phi})^T\phi(\overline{\mathbf{z}}(\mathbf{x}_{hj}, h)) \\ &= \frac{1}{2}\sum_{t=1}^{K}\sum_{i=1}^{|S_t|}\alpha_{ti}\phi(\overline{\mathbf{z}}(\mathbf{x}_{ti}, t))^T\phi(\overline{\mathbf{z}}(\mathbf{x}_{hj}, h)) \\ &= \frac{1}{2}\sum_{t=1}^{K}\sum_{i=1}^{|S_t|}\alpha_{ti}K(\mathbf{z}(\mathbf{x}_{ti}, t) + \Delta\overline{\mathbf{z}}(\mathbf{x}_{ti}, t), \mathbf{z}(\mathbf{x}_{hj}, h) + \Delta\mathbf{z}(\mathbf{x}_{hj}, h)) \\ &= \frac{1}{2}\sum_{t=1}^{K}\sum_{i=1}^{|S_t|}\alpha_{ti}K(\mathbf{z}(\mathbf{x}_{ti}, t) + \Delta\overline{\mathbf{z}}(\mathbf{x}_{ti}, t), \mathbf{z}(\mathbf{x}_{hj}, h)) \\ &\quad + \frac{1}{2}\Delta\mathbf{z}(\mathbf{x}_{hj}, h)^T\sum_{t=1}^{K}\sum_{i=1}^{|S_t|}\alpha_{ti}K'(\mathbf{z}(\mathbf{x}_{ti}, t) + \Delta\overline{\mathbf{z}}(\mathbf{x}_{ti}, t), \mathbf{z}(\mathbf{x}_{hj}, h)) \end{aligned} \tag{67}$$

By utilizing the Cauchy–Schwarz inequality, the optimal value of $\Delta\mathbf{x}_{hj}$ is

$$\Delta\mathbf{z}(\mathbf{x}_{hj}, h) = \delta_{hj}\frac{\widetilde{\mathbf{u}}_{hj}}{||\widetilde{\mathbf{u}}_{hj}||}, \quad j = 1, \ldots, |S_h|, \ \ h = 1, \ldots, K \tag{68}$$

where it has

$$\widetilde{\mathbf{u}}_{hj} = \sum_{t=1}^{K} \sum_{i=1}^{|S_t|} \alpha_{ti} K' \left(\mathbf{z}(\mathbf{x}_{ti}, t) + \Delta \overline{\mathbf{z}}(\mathbf{x}_{ti}, t), \mathbf{z}(\mathbf{x}_{hj}, h) \right).$$

□

References

1. Schölkopf B, Williamson RC, Smola A, Shawe-Taylor J (1999) Support vector method for novelty detection. In: Proceedings of neural information processing systems 1999, pp 582–588
2. Manevitz LM, Yousef M (2002) One-class SVMs for document classiffication. J Mach Learn Res 2:139–154
3. Ma J, Perkins S (2003) Time-series novelty detection using one-class support vector machines. In: Proceedings of international joint conference on neural networks 2003, pp 1741–1745
4. Li J, Su L, Cheng C (2011) Finding pre-images via evolution strategies. Appl Soft Comput 11(6):4183–4194
5. Takruri M, Rajasegarar S, Challa S, Leckie C, Palaniswami M (2011) Spatio-temporal modelling-based drift-aware wireless sensor networks. Wirel Sens Syst 1(2):110–122
6. Münoz-Marí J, Bovolo F, Gomez-Chova L, Bruzzone L, Camp-Valls G (2010) Semisupervised one-class support vector machines for classification of remote sensing data. IEEE Trans Geosci Remote Sens 48(8):3188–3197
7. Yu H, Han J, Chang KCC (2004) Pebl: web page classification without negative examples. IEEE Trans Knowl Data Eng 16(1):70–81
8. Fung GPC, Yu JX, Lu H, Yu PS (2006) Text classification without negative examples revisit. IEEE Trans Knowl Data Eng 18:6–20
9. Liu B, Xiao Y, Cao L, Yu PS (1995) One-class-based uncertain data stream learning. In: Proceedings of SIAM international conference on data mining 2011, pp 992–1003
10. Pan SJ, Tsand IW, Kwok JT, Yang Q (2011) Domain adaptation via transfer component analysis. IEEE Trans Neural Netw 22(2):199–210
11. Aggarwal CC, Yu PS (2009) A survey of uncertain data algorithms and applications. IEEE Trans Knowl Data Eng 21(5):609–623
12. Kriegel HP, Pfeifle M (2005) Hierarchical density based clustering of uncertain data. In: Proceedings of international conference on data engineering 2005, pp 689–692
13. Ngai W, Kao B, Chui C, Cheng R, Chau M, Yip KY (2006) Efficient clustering of uncertain data. In: Proceedings of international conference on data mining 2006, pp 436–445
14. Aggarwal CC (2007) On density based transforms for uncertain data mining. In: Proceedings of international conference on data engineering 2007, pp 866–875
15. Bi J, Zhang T (2004) Support vector classification with input data uncertainty. In: Proceedings of neural information processing systems, 2004
16. Gao C, Wang J (2010) Direct mining of discriminative patterns for classifying uncertain data. In: Proceedings of ACM SIGKDD conference on knowledge discovery and data mining 2010, pp 861–870
17. Tsang S, Kao B, Yip KY, Ho WS, Lee SD (2011) Decision trees for uncertain data. IEEE Trans Knowl Data Eng 23(1):64–78
18. Murthy R, Ikeda R, Widom J (2011) Making aggregation work in uncertain and probabilistic databases. IEEE Trans Knowl Data Eng 22(8):1261–1273
19. Yuen SM, Tao Y, Xiao X, Pei J, Zhang D (2010) Superseding nearest neighbor search on uncertain spatial databases. IEEE Trans Knowl Data Eng 22(7):1041–1055
20. Sun L, Cheng R, Cheung DW, Cheng J (2010) Mining uncertain data with probabilistic guarantees. In: Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining 2010, pp 273–282
21. Dai W, Xue G, Yang Q, Yu Y (2007) Transferring naive bayes classifiers for text classification. In: Proceedings of the AAAI conference on artificial intelligence 2007, pp 540–545
22. Jiang J, Zhai C (2007) Instance weighting for domain adaptation in NLP. In: Proceedings of the association for computational linguistics 2007, pp 264–271
23. Liao X, Xue Y, Carin L (2005) Logistic regression with an auxiliary data source. In: Proceedings of the international conference on machine learning 2005, pp 505–512
24. Huang J, Smola A, Gretton A, Borgwardt KM, Schölkopf B (2007) Correcting sample selection bias by unlabeled data. In: Proceedings of the neural information processing systems 2007, pp 601–608

25. Zheng VW, Yang Q, Xiang W, Shen D (2008) Transferring localization models over time. In: Proceedings of the AAAI conference on artificial intelligence 2008, pp 1421–1426
26. Pan SJ, Shen D, Yang Q, Kwok JT (2008) Transferring localization models across space. In: Proceedings of the AAAI conference on artificial Intelligence 2008, pp 1383–1388
27. Raykar VC, Krishnapuram B, Bi J, Dundar M, Rao RB (2008) Bayesian multiple instance learning: automatic feature selection and inductive transfer. In: Proceedings of the international conference on machine learning 2008, pp 808–815
28. Pan SJ, Qiang Y (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
29. Dai W, Yang Q, Xue G, Yu Y (2007) Boosting for transfer learning. In: Proceedings of the international conference on machine learning 2007, pp 193–200
30. Raina R, Battle A, Lee H, Packer B, Ng AY (2007) Self-taught learning: transfer learning from unlabeled data. In: Proceedings of the international conference on machine learning 2007, pp 759–766
31. Dai W, Xue G, Yang Q, Yu Y (2007) Co-clustering based classification for out-of-domain documents. In: Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining 2007, pp 432–444
32. Ando RK, Zhang T (2005) A high-performance semi-supervised learning method for text chunking. In: Proceedings of the association for computational linguistics 2005, pp 1–9
33. Lawrence ND, Platt JC (2004) Learning to learn with the informative vector machine. In: Proceedings of the international conference on machine learning 2004, pp 432–444
34. Schwaighofer A, Tresp V, Yu K (2005) Learning gaussian process kernels via hierarchical bayes. In: Proceedings of the neural information processing systems 2005, pp 1209–1216
35. Gao J, Fan W, Jiang J, Han J (2008) Knowledge transfer via multiple model local structure mapping. In: Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining 2008, pp 283–291
36. Mihalkova L, Huynh T, Mooney RJ (2007) Mapping and revising markov logic networks for transfer learning. In: Proceedings of the AAAI conference on artificial intelligence 2007, pp 608–614
37. Mihalkova L, Mooney RJ (2008) Transfer learning by mapping with minimal target data. In: Proceedings of workshop transfer learning for complex tasks with AAAI, 2008
38. Davis J, Domingos P (2008) Deep transfer via second-order markov logic. In: Proceedings of workshop transfer learning for complex tasks with AAAI, 2008
39. Bonilla EV, Agakov F, Williams C (2007) Kernel multi-task learning using task-specific features. In: Proceedings of the international conference on artificial intelligence and statistics 2007, pp 43–50
40. Yu K, Tresp V, Schwaighofer A (2005) Learning gaussian processes from multiple tasks. In: Proceedings of the international conference on machine learning 2005, pp 1012–1019
41. Bakker B, Heskes T (2003) Task clustering and gating for bayesian multitask learning. J Mach Learn Res 4:83–99
42. Huffel SV, Vandewalle J (1991) The total least squares problem: computational aspects and analysis. Frontiers in applied mathematics. SIAM Press, Philadelphia
43. Vapnik V (1998) Statistical learning theory. Frontiers in applied mathematics. Springer, London
44. Wang F, Zhao B, Zhang CS (2010) Linear time maximum margin clustering. IEEE Trans Neural Netw 21(2):319–332
45. Chen J, Liu X (2014) Transfer learning with one-class data. Pattern Recognit Lett 37(1):32–40
46. Schölkopf B, Herbrich R, Smola AJ, Williamson RC (2001) A generalized representer theorem. In: Proceedings of the annual conference on learning theory 2001, pp 416–426
47. Chang C-C, Lin C-J (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):1–27
48. William J, Shaw M (1986) On the foundation of evaluation. Am Soc Inf Sci 37(5):346–348
49. Tax DMJ, Duin RPW (2004) Support vector data description. Mach Learn 54(1):45–66
50. Cao B, Pan J, Zhang Y, Yeung DY, Yang Q (2010) Adaptive transfer learning. In: Proceedings of the AAAI conference on artificial intelligence, 2010
51. Aggarwal CC, Yu PS (2008) A framework for clustering uncertain data streams. In: Proceedings of the international conference on data engineering 2008, pp 150–159
52. Cole R, Fanty MA (1990) Spoken letter recognition. In: Proceedings of the workshop on speech and natural language 1990, pp 385–390
53. Yin J, Yang Q, Pan JJ (2008) Sensor-based abnormal human-activity detection. IEEE Trans Knowl Data Eng 20(8):1082–1090
54. Tsang IW, Kwok JT, Cheung PM (2005) Core vector machines: Fast SVM training on very large data sets. J Mach Learn Res 6:363–392
55. Dong JX, Devroye L, Suen CY (2005) Core vector machines: fast SVM training algorithm with decomposition on very large data sets. IEEE Trans Pattern Anal Mach Intell 27(4):603–618

56. Tresp V (2000) A Bayesian committee machine. Neural Comput 12(11):2719–2741
57. Shalev-Shwartz S, Singer Y, Srebro N (2007) Pegasos: primal estimated sub-gradient solver for SVM. In: Proceedings of the international conference on machine learning 2007, pp 807–814
58. Kivinen J, Smola AJ, Williamson RC (2004) Online learning with kernels. IEEE Trans Signal Process 52(8):1–12
59. Dragomir SS (2003) A survey on cauchy-bunyakovsky-schwarz type discrete inequalities. J Inequal Pure Appl Math 4(3):1–142

Yanshan Xiao received the Ph.D. degree in computer science from the Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia, in 2011. She is with the Faculty of Computer, Guangdong University of Technology. Her research interests include multiple-instance learning, support vector machine, data mining and machine learning.

Bo Liu is with the Faculty of Automation, Guangdong University of Technology. His research interests include machine learning and data mining. He has published papers on IEEE Transactions on Neural Networks, IEEE Transactions on Knowledge and Data Engineering, Knowledge and Information Systems, International Joint Conferences on Artificial Intelligence (IJCAI), IEEE International Conference on Data Mining (ICDM), SIAM International Conference on Data Mining (SDM) and ACM International Conference on Information and Knowledge Management (CIKM).

Philip S. Yu received the B.S. Degree in E.E. from National Taiwan University, the M.S. and Ph.D. degrees in E.E. from Stanford University, and the MBA degree from New York University. He is a Professor in the Department of Computer Science at the University of Illinois at Chicago and also holds the Wexler Chair in Information Technology. He spent most of his career at IBM Thomas J. Watson Research Center and was manager of the Software Tools and Techniques group. His research interests include data mining, privacy preserving data publishing, data stream, Internet applications and technologies, and database systems. Dr. Yu has published more than 710 papers in refereed journals and conferences. He holds or has applied for more than 300 US patents. Dr. Yu is a Fellow of the ACM and the IEEE. He is the Editor-in-Chief of ACM Transactions on Knowledge Discovery from Data. He is on the steering committee of the IEEE Conference on Data Mining and ACM Conference on Information and Knowledge Management and was a member of the IEEE Data Engineering steering committee. He was the Editor-in-Chief of IEEE Transactions on Knowledge and Data Engineering (2001–2004). He had also served as an associate editor of ACM Transactions on the Internet Technology (2000–2010) and Knowledge and Information Systems (1998–2004). In addition to serving as program committee member on various conferences, he was the program chair or co-chairs of the 2009 IEEE Intl. Conf. on Service-Oriented Computing and Applications, the IEEE Workshop of Scalable Stream Processing Systems (SSPS07), the IEEE Workshop on Mining Evolving and Streaming Data (2006), the 2006 joint conferences of the 8th IEEE Conference on E-Commerce Technology (CEC 06) and the 3rd IEEE Conference on Enterprise Computing, E-Commerce and E-Services (EEE 06), the 11th IEEE Intl. Conference on Data Engineering, the 6th Pacific Area Conference on Knowledge Discovery and Data Mining, the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, the 2nd IEEE Intl. Workshop on Research Issues on Data Engineering: Transaction and Query Processing, the PAKDD Workshop on Knowledge Discovery from Advanced Databases, and the 2nd IEEE Intl. Workshop on Advanced Issues of E-Commerce and Web-based Information Systems. He served as the general chair or co-chairs of the 2009 IEEE Intl. Conf. on Data Mining, the 2009 IEEE Intl. Conf. on Data Engineering, the 2006 ACM Conference on Information and Knowledge Management, the 1998 IEEE Intl. Conference on Data Engineering, and the 2nd IEEE Intl. Conference on Data Mining. He had received several IBM honors including 2 IBM Outstanding Innovation Awards, an Outstanding Technical Achievement Award, 2 Research Division Awards and the 94th plateau of Invention Achievement Awards. He was an IBM Master Inventor. Dr. Yu received a Research Contributions Award from IEEE Intl. Conference on Data Mining in 2003 and also an IEEE Region 1 Award for promoting and perpetuating numerous new electrical engineering concepts in 1999.

Zhifeng Hao received the B.S. degree from Sun Yat-Sen University, Guangzhou, China, and the Ph.D. degree in mathematics from Nanjing University, Nanjing, China, in 1990, and 1995, respectively. He is currently a Professor with the Faculty of Computer Science, Guangdong University of Technology and School of Computer Science and Engineering, South China University of Technology, Guangzhou. His current research interests include algebra, machine learning, data mining, and evolutionary algorithms.