Information Sciences 349–350 (2016) 216–228

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Semi-supervised classification method through oversampling and common hidden space

Aimei Dong[a,c], Fu-lai Chung[b], Shitong Wang[a,b,*]

[a] School of Digital Media, Jiangnan University, Wuxi, JiangSu, China
[b] Department of Computing, Hong Kong Polytechnic University, Hong Kong, China
[c] School of Information, Qilu University of Technology, Jinan, ShanDong, China

ARTICLE INFO

Article history:
Received 17 July 2015
Revised 15 February 2016
Accepted 21 February 2016
Available online 27 February 2016

Keywords :
Semi-supervised classification
Oversampling
Common hidden space
Dimensionality augmentation

ABSTRACT

Semi-supervised classification methods attempt to improve classification performance based on a small amount of labeled data through full use of abundant unlabeled data. Although existing semi-supervised classification methods have exhibited promising results in many applications, they still have drawbacks, including performance degeneration, due to the introduction of unlabeled data and partially false labels in a small amount of labeled data. To circumvent such drawbacks, a new semi-supervised classification method OCHS-SSC through oversampling and a common hidden space is proposed in the paper. The primary characteristics of the proposed method include two aspects. One is that unlabeled data are only used to generate new synthetic data to extend the minimal amount of labeled data. The other is that the final classifier is learned in the extended feature space, which is composed of the original feature space and the common hidden space found between labeled data and the synthetic data instead of the original feature space. Extensive experiments on 23 datasets indicate the effectiveness of the proposed method.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Pattern classification is a research technique in machine learning and has been applied to many related fields, such as web-page classification and spam filtering. There are an increasing number of applications [4,12,19,22,23,28,29] with a great deal of unlabeled training data and a small amount of labeled data. Abundant unlabeled data are quite easy to acquire, whereas a small amount of labeled data is usually costly and difficult to acquire. To solve these problems, semi-supervised learning [5,24,36,38] is proposed to learn an effective pattern classifier from a small amount of labeled data with the aid of abundant unlabeled data.

To date, many semi-supervised classification methods have been developed using different approaches. The most distinguished achievements among all of the semi-supervised classification methods include (1) semi-supervised support vector machines [15], which maximize the margin of labeled and unlabeled data through adaption of the hyperplane of SVM and the labels of unlabeled data; (2) generative methods [20–21], in which all of the training data (labeled and/or unlabeled) are assumed to be generated from the same generation model and its parameters. As the bridge linking labeled and unlabeled data, the generation model and its parameters are estimated through maximizing the corresponding likelihood with EM-based algorithms. It is important but difficult for such methods to find an appropriate model. If the training data are

[*] Corresponding author at: School of Digital Media, Jiangnan University, Wuxi, JiangSu, China. Tel.: +86 13182791468.
E-mail address: wxwangst@aliyun.com (S. Wang).

http://dx.doi.org/10.1016/j.ins.2016.02.042
0020-0255/© 2016 Elsevier Inc. All rights reserved.

not subject to the hypothetical model, unlabeled data will become harmful for the final classifier; (3) graph-based methods [3,18,33,37], in which the training data are mapped into a graph embodying the relationship among the training data. The label information is then propagated based on the graph. If the constructed graph is not consistent with the inherent law of the training data, the introduction of abundant unlabeled data will be dangerous for the final classifier; and (4) self-labeled [26–27] methods, which accept that their own predictions tend to be correct in an iterative process through different mechanisms such as self-training [16,32], tri-training [34] or disagreement-based models [35]. If unlabeled data are falsely predicted during the iterative self-labeling process, harmful results will perhaps occur. Thus, how to control false predictions for unlabeled data is critical for such methods.

All of these methods have the same goal of taking full advantage of unlabeled data. Although semi-supervised classification methods have shown promising performance in many applications, several scholars [2,25] have found unavoidable disadvantages immanent to these methods. In other words, the performance of the final classifier might degrade due to the introduction of unlabeled data. This discovery inspires us to consider how to use unlabeled data carefully. On this topic, Zhou [17] and Chen [30] proceeded by developing the S4VM and SA-SSCCM methods, respectively. Both authors advance safe strategies for using unlabeled data in unique ways and obtained outstanding performance. The common idea of the two approaches is to use labeled and unlabeled data directly. When labeled data are partially falsely labeled and/or unlabeled data contain outliers, the performance of these methods might degrade more seriously.

This paper focuses on solving semi-supervised classification problems with the characteristic of partially false labels of labeled data and the presence of outliers among unlabeled data. The overall approach to addressing these problems contains two aspects. One is not to use unlabeled data directly but rather to generate new synthetic data with the help of unlabeled data. The other is not to learn the final classifier in the original feature space but rather in the extended feature space. The latter is composed of the original feature space and the augmented feature space, which is shared by both the small amount of labeled data and the synthetic data. To this end, a new semi-supervised classification method OCHS-SSC through oversampling and the common hidden space is proposed in this paper. Extensive experiments on six benchmark datasets and seventeen UCI datasets have confirmed the efficiency of the proposed method.

The remainder of this paper is organized as follows. The preliminaries of the proposed method are introduced in Section 2. Section 3 describes the proposed method. Extensive experimental results are provided in Section 4 to validate the effectiveness of the proposed method. In the last section, the conclusions are provided, and future works are discussed.

2. Preliminaries of the proposed method

2.1. Definitions of semi-supervised classification

In a semi-supervised classification scenario, all of the data can be split into the labeled part and the unlabeled part. Let the labeled dataset be $L = \{\boldsymbol{x}_i, y_i\}_{i=1}^{n_l}$ and the unlabeled dataset be $U = \{\boldsymbol{x}_j\}_{j=1}^{n_u}$, in which $\boldsymbol{x}_i \in \mathbb{R}^{\mathrm{d}}$ and $\boldsymbol{x}_j \in \mathbb{R}^{\mathrm{d}}$, $y_i \in \{+1, -1\}$, and $n_u \gg n_l$. The goal of semi-supervised classification is to learn a better classifier for future unseen data (inductive semi-supervised learning) or for the unlabeled data (transductive semi-supervised learning) [7]. In this study, we perform experiments from the perspective of both inductive semi-supervised learning and transductive semi-supervised learning.

2.2. Motivation for using oversampling technology

In traditional semi-supervised classification methods, unlabeled data are self-labeled by using certain mechanisms such as manifold assumption and graph-based models to expand the labeled dataset. In a self-labeling process, there might be wrong unlabeled data included in the labeled dataset. The condition occurs due to the following:

(1) There might be noise in the unlabeled dataset. The existence of noise in the unlabeled dataset can easily cause the wrong unlabeled examples to be added to the labeled dataset.
(2) The size of the labeled dataset is so small that the labeled dataset cannot reflect its genuine data distribution. The shortage of labeled data will cause wrong unlabeled examples to be added to the labeled dataset. This problem is more obvious when some labeled data are very close to the decision boundary.

The first reason shows that it is better not to use the unlabeled dataset directly in semi-supervised classification; the second reason shows that it is necessary to expand the labeled dataset. Considering these points, the usage of oversampling technology for the labeled dataset based on the labeled dataset and the unlabeled dataset is a good choice in semi-supervised classification.

2.3. Motivation for using the common hidden space

In traditional semi-supervised classification methods, the unlabeled dataset is labeled in a repeated process until a certain criterion is satisfied with the direct usage of the labeled dataset. In the repeated labeling process, if labeled data are attacked by certain external factors and some of the labels of the labeled dataset are consequently wrong, then more wrong unlabeled data are included into the training set. To overcome this disadvantage, a common hidden space between the labeled dataset and the generated synthetic dataset is found such that the proposed method in this study is performed

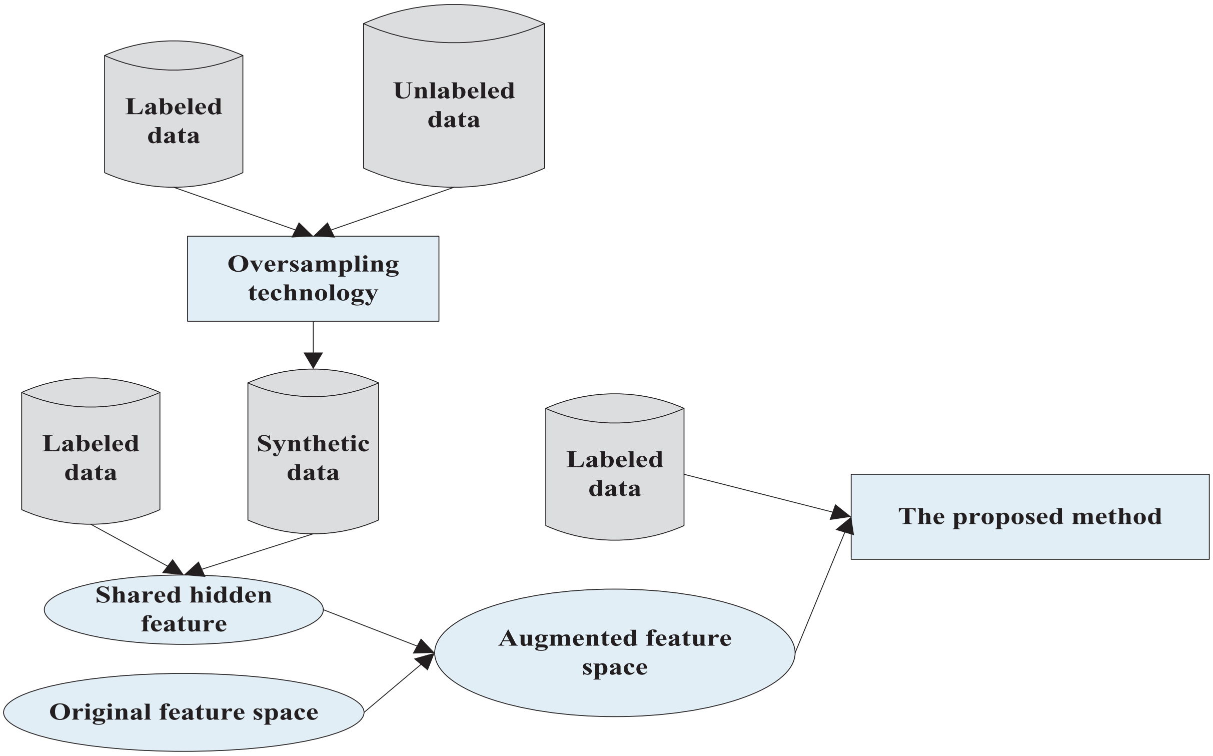


Fig. 1. Principle of the proposed method.

not on the original feature space but on the augmented feature space composed of the original and the common hidden feature spaces for labeled data.

2.4. Principle of the proposed method

In summary, for the proposed method, the unlabeled dataset is only used to generate the synthetic dataset and expand the labeled dataset, and the labeled dataset is not used on the original feature space but on the augmented feature space. The advantages of such a mechanism involve two aspects: (1) removing the dependence of the unlabeled dataset and (2) removing the dependence of the labels of the labeled dataset. Note that the idea of the augmented feature space can be supported by the conclusion made in [39], i.e., *for the learning of a classifier, the negative influence caused by errors in features is far less than that caused by errors in labels.*

Three steps are taken to obtain the final classifier in the proposed method. (1) The oversampling technology is adopted to generate the synthetic dataset based on all of the data including labeled and unlabeled data. (2) A common hidden space is found between the labeled dataset and the generated synthetic dataset. Finally, (3) the final classifier is learned by the labeled dataset on the augmented feature space, which is composed of the original feature space and the common hidden feature space. The mechanism of the proposed method can be depicted as in Fig. 1.

3. The proposed method

3.1. Problem formulation

For labeled dataset L and unlabeled dataset U, let $S = \{\boldsymbol{x}_i, y_i\}_{i=1}^{n_s}$ denote the generated synthetic dataset, in which $\boldsymbol{x}_i \in \mathbb{R}^{\mathrm{d}}$, $y_i \in \{+1, -1\}$. The common hidden space can be found by the following strategy.

For the labeled dataset L and the generated synthetic dataset S, the Parzen Window-based probability density distributions can be respectively expressed as follows:

$$\widehat{P}_L(\boldsymbol{x}) = \frac{1}{n_l \times \sqrt{2\pi}\,\sigma} \sum_{i=1}^{n_l} e^{-\frac{\left\|\boldsymbol{x}-\boldsymbol{x}_i^L\right\|^2}{2\sigma^2}} \tag{1}$$

$$\widehat{P}_S(\boldsymbol{x}) = \frac{1}{n_s \times \sqrt{2\pi}\,\sigma} \sum_{j=1}^{n_s} e^{-\frac{\left\|\boldsymbol{x}-\boldsymbol{x}_j^S\right\|^2}{2\sigma^2}} \tag{2}$$

where $\sigma > 0$ denotes the window width used by both density distributions.

Let $\boldsymbol{\Theta} \in \mathbb{R}^{r\times d}$ denote an orthonormal matrix with $\boldsymbol{\Theta}\boldsymbol{\Theta}^T = \mathbf{I}_{r\times r}$, and assume that there exists a hidden feature space $\boldsymbol{\Theta x} \in \mathbb{R}^r$. For a classification task, it is necessary to find a projection in the corresponding hidden feature space to obtain an ideal classification plane. Let us denote the projective vector as $\boldsymbol{v} \in \mathbb{R}^r$, and let $\tilde{\boldsymbol{x}} = \boldsymbol{\Theta x}$. The projective density distributions of L and S in the hidden feature space can be respectively expressed as follows:

$$\widehat{P}_L(\boldsymbol{v}^T\tilde{\boldsymbol{x}}) = \widehat{P}_L(\boldsymbol{v}^T\boldsymbol{\Theta x}) = \frac{1}{n_l \times \sqrt{2\pi\sigma}}\sum_{i=1}^{n_l} e^{-\frac{\boldsymbol{v}^T\boldsymbol{\Theta}(\boldsymbol{x}-\boldsymbol{x}_i^L)(\boldsymbol{x}-\boldsymbol{x}_i^L)^T\boldsymbol{\Theta}^T\boldsymbol{v}}{2\sigma^2}} \tag{3}$$

$$\widehat{P}_S(\boldsymbol{v}^T\tilde{\boldsymbol{x}}) = \widehat{P}_S(\boldsymbol{v}^T\boldsymbol{\Theta x}) = \frac{1}{n_s \times \sqrt{2\pi\sigma}}\sum_{j=1}^{n_s} e^{-\frac{\boldsymbol{v}^T\boldsymbol{\Theta}(\boldsymbol{x}-\boldsymbol{x}_j^S)(\boldsymbol{x}-\boldsymbol{x}_j^S)^T\boldsymbol{\Theta}^T\boldsymbol{v}}{2\sigma^2}} \tag{4}$$

In this study, we take the following integrated squared error between $\widehat{P}_L(\boldsymbol{v}^T\tilde{\boldsymbol{x}})$ and $\widehat{P}_S(\boldsymbol{v}^T\tilde{\boldsymbol{x}})$ to measure the difference among these two density distributions, i.e.,

$$J_1 = \int \left(\widehat{P}_L(\boldsymbol{v}^T\tilde{\boldsymbol{x}}) - \widehat{P}_S(\boldsymbol{v}^T\tilde{\boldsymbol{x}}))\right)^2 dx \tag{5}$$

By minimizing J_1, it is expected that the labeled and unlabeled datasets have the maximal commonality in the projective hidden feature space.

Let $G(\boldsymbol{v}^T\tilde{\boldsymbol{x}}, \boldsymbol{v}^T\tilde{\boldsymbol{x}}_i, \sigma^2) = G(\boldsymbol{v}^T\boldsymbol{\Theta x}, \boldsymbol{v}^T\boldsymbol{\Theta x}_i, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{\boldsymbol{v}^T\boldsymbol{\Theta}(\boldsymbol{x}-\boldsymbol{x}_i)(\boldsymbol{x}-\boldsymbol{x}_i)^T\boldsymbol{\Theta}^T\boldsymbol{v}}{2\sigma^2}}$; then $\widehat{P}_L(\boldsymbol{v}^T\tilde{\boldsymbol{x}})$ and $\widehat{P}_S(\boldsymbol{v}^T\tilde{\boldsymbol{x}})$ can be denoted as follows:

$$\widehat{P}_L(\boldsymbol{v}^T\tilde{\boldsymbol{x}}) = \frac{1}{n_l}\sum_{i=1}^{n_l} G(\boldsymbol{v}^T\tilde{\boldsymbol{x}}, \boldsymbol{v}^T\tilde{\boldsymbol{x}}_i^L, \sigma^2) \tag{6}$$

$$\widehat{P}_S(\boldsymbol{v}^T\tilde{\boldsymbol{x}}) = \frac{1}{n_s}\sum_{j=1}^{n_s} G(\boldsymbol{v}^T\tilde{\boldsymbol{x}}, \boldsymbol{v}^T\tilde{\boldsymbol{x}}_j^S, \sigma^2) \tag{7}$$

Thus, we have $J = \int \widehat{P}_L^2(\boldsymbol{v}^T\tilde{\boldsymbol{x}})dx - 2\int \widehat{P}_L(\boldsymbol{v}^T\tilde{\boldsymbol{x}})\widehat{P}_S(\boldsymbol{v}^T\tilde{\boldsymbol{x}})dx + \int \widehat{P}_S^2(\boldsymbol{v}^T\tilde{\boldsymbol{x}})dx$. Because $\int \widehat{P}_L^2(\boldsymbol{v}^T\tilde{\boldsymbol{x}})dx = \int \widehat{P}_L(\boldsymbol{v}^T\tilde{\boldsymbol{x}})\widehat{P}_L(\boldsymbol{v}^T\tilde{\boldsymbol{x}})dx$, which actually denotes the mathematical expectation, i.e., $E[\widehat{P}_L(\boldsymbol{v}^T\tilde{\boldsymbol{x}})]$ of $\widehat{P}_L(\boldsymbol{v}^T\tilde{\boldsymbol{x}})$, without any prior acknowledge about labeled dataset L, we can roughly approximate the sum of $\widehat{P}_L(\boldsymbol{v}^T\tilde{\boldsymbol{x}})$ on L as 1; therefore, $E[\widehat{P}_L(\boldsymbol{v}^T\tilde{\boldsymbol{x}})]$ can be approximately equivalent to $1/n_l$. Similarly, $\int \widehat{P}_S^2(\boldsymbol{v}^T\tilde{\boldsymbol{x}})dx$ can be approximated as $1/n_s$. Because $\int G(\boldsymbol{x}, \boldsymbol{x}_i, \sigma_1^2)G(\boldsymbol{x}, \boldsymbol{x}_j, \sigma_2^2)d\boldsymbol{x} = G(\boldsymbol{x}_i, \boldsymbol{x}_j, \sigma_1^2 + \sigma_2^2)$ [10], we have $\int \widehat{P}_L(\boldsymbol{v}^T\tilde{\boldsymbol{x}})\widehat{P}_S(\boldsymbol{v}^T\tilde{\boldsymbol{x}})dx = \frac{1}{n_l \times n_s}\sum_{i=1}^{n_l}\sum_{j=1}^{n_s} G(\boldsymbol{v}^T\tilde{\boldsymbol{x}}_i^L, \boldsymbol{v}^T\tilde{\boldsymbol{x}}_j^S, 2\sigma^2)$. Thus, we can approximate J_1 as $J_1 \approx \frac{1}{n_l} + \frac{1}{n_s} - \frac{2}{n_l \times n_s}\sum_{i=1}^{n_l}\sum_{j=1}^{n_s} G(\boldsymbol{v}^T\tilde{\boldsymbol{x}}_i^L, \boldsymbol{v}^T\tilde{\boldsymbol{x}}_j^S, 2\sigma^2)$. In other words,

$$\begin{aligned} &\arg\min_{\boldsymbol{v},\boldsymbol{\Theta}} J_1 \approx \arg\max_{\boldsymbol{v},\boldsymbol{\Theta}} \sum_{i=1}^{n_l}\sum_{j=1}^{n_s} G(\boldsymbol{v}^T\boldsymbol{\Theta x}_i^L, \boldsymbol{v}^T\boldsymbol{\Theta x}_j^S, 2\sigma^2) \\ &s.t. \boldsymbol{\Theta}\boldsymbol{\Theta}^T = \mathbf{I}_{r\times r} \end{aligned} \tag{8}$$

Because Eq. (8) is very complicated and difficult to solve, it might be approximated by using a simpler formulation based on its Taylor's expansion [11]. According to its Taylor's expansion, we have

$$G(\boldsymbol{v}^T\boldsymbol{\Theta x}_i^L, \boldsymbol{v}^T\boldsymbol{\Theta x}_j^S, 2\sigma^2) \approx \frac{1}{\sqrt{2\pi\sigma}}\left(1 - \frac{1}{4\sigma^2}\boldsymbol{v}^T\boldsymbol{\Theta}(\boldsymbol{x}_i^L - \boldsymbol{x}_j^S)(\boldsymbol{x}_i^L - \boldsymbol{x}_j^S)^T\boldsymbol{\Theta}^T\boldsymbol{v}\right) \tag{9}$$

Thus, Eq. (8) can be approximated as

$$\begin{aligned} &\arg\min_{\boldsymbol{v},\boldsymbol{\Theta}} \boldsymbol{v}^T\boldsymbol{\Theta}\sum_{i=1}^{n_l}\sum_{j=1}^{n_s}(\boldsymbol{x}_i^L - \boldsymbol{x}_j^S)(\boldsymbol{x}_i^L - \boldsymbol{x}_j^S)^T\boldsymbol{\Theta}^T\boldsymbol{v} \\ &s.t.\ \boldsymbol{\Theta}\boldsymbol{\Theta}^T = \mathbf{I}_{r\times r} \end{aligned} \tag{10}$$

In this study, Eq. (10) will be used to find the optimal orthonormal matrix and the projective vector in the corresponding hidden feature space.

If SVM is used, the above mechanism can be adopted to leverage the useful information in the generated synthetic dataset for the training of SVM. In particular, the following classification plane is suggested for the proposed classifier:

$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{v}^T\tilde{\boldsymbol{x}} = \boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{v}^T\boldsymbol{\Theta x} = (\boldsymbol{w}^T + \boldsymbol{v}^T\boldsymbol{\Theta})\boldsymbol{x} \tag{11}$$

where $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ denote the feature vectors in the original and common hidden feature spaces, respectively; $\boldsymbol{w}$ and $\boldsymbol{v}$ denote the projective vectors in the original and common hidden feature spaces, respectively.

Table 1
Algorithm SDG.

```
Algorithm SDG(L, U, f, k) (Synthetic Dataset Generation)

Input: The labeled dataset $L$ and the unlabeled dataset $U$, oversampling factor $f$, number of neighbors $k$
output: The synthetic dataset $S$
Initialization: $S = \oslash$
for $i$=1 to Number of Classes do
    perclass[i]=getFromClass(L,i)
    n[i]=Number of perclass[i];
    s[i]=ceil(n[i]* f)
    randomize perclass[i]
    for j=1 to s[i] do
        neighbors [1..k]=compute k nearest neighbors for perclass[i][j] in the unlabeled dataset U
        nn=random number between 1 and k
        sample=perclass[i][j];
        Nearest= U[neighbors[nn]]
        for d=1 to Number of Attributes do
            dif=Nearset[d]-sample[d];
            gap=Random number between 0 and 1
            synthetic[d]=sample[d]+gap*dif
        end for
        $S = S \cup synthetic$
    end for
end for
return S
```

Finally, based on the large margin principle of the classical SVM and the minimum integrated squared error between the probability distributions of the labeled samples and the generated synthetic samples in Eq. (10), the following objective function is proposed for the classifier in Eq. (11).

$$\begin{aligned}
&\arg\min_{\boldsymbol{w},\boldsymbol{v},\boldsymbol{\Theta},\xi_i} J_2 = \arg\min_{\boldsymbol{w},\boldsymbol{v},\Theta,\xi_i} \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\lambda_1}{2}\left\|\boldsymbol{w}+\boldsymbol{\Theta}^T\boldsymbol{v}\right\|^2 + C\sum_{i=1}^{n_l}\xi_i + \frac{\lambda_2}{2}\boldsymbol{v}^T\boldsymbol{\Theta}\sum_{i=1}^{n_l}\sum_{j=1}^{n_s}(\boldsymbol{x}_i^L-\boldsymbol{x}_j^S)(\boldsymbol{x}_i^L-\boldsymbol{x}_j^S)^T\boldsymbol{\Theta}^T\boldsymbol{v}\\
&s.t.\ y_i(\boldsymbol{w}^T+\boldsymbol{v}^T\boldsymbol{\Theta})\boldsymbol{x}_i \geq 1-\xi_i\ \xi_i \geq 0\ i=1,2,\ldots,n_l\\
&\boldsymbol{\Theta}\boldsymbol{\Theta}^T = \mathbf{I}_{r\times r}
\end{aligned} \tag{12}$$

3.2. Solution

3.2.1. Generation of the synthetic dataset

To achieve a balance between the sizes of both classes in an imbalanced data classification task in which the majority class owns many more samples than does the other (i.e., the minority class), an oversampling technique is used to generate new synthetic data for the minority class. At present, two typical oversampling methods are SMOTE [6], in which synthetic samples are generated to be nearby the data samples in the minority class, and Borderline-SMOTE [14], in which synthetic samples are generated within appropriate zones in the minority class. Similar to the study lines in [6,14], to generate new synthetic data, we perform certain operations on the labeled dataset and the unlabeled dataset in the proposed method. The procedure of generating the synthetic dataset can be summarized as algorithm SDG (synthetic dataset generation), as shown in Table 1. Note that algorithm SDG is a modified version of the SMOTE method in [6]. The SMOTE method was originally designed for an imbalanced dataset and limited to oversampling the minority class. In the SDG algorithm, we use the potential idea of SMOTE to generate new samples of all classes. For consistency with the distribution of the labeled dataset, the amount of synthetic data from different classes is proportional to the amount of labeled data from the corresponding class. In other words, if the ratio of negative to positive in the labeled dataset is *m*, then that ratio in the synthetic dataset should also be *m*.

Note that a similar idea can also be seen in [27]. However, there exist two differences between them. (1) In [27], the synthetic data are generated from the dataset, which is composed of labeled and unlabeled data. However, in the proposed method, the synthetic data are only generated from the unlabeled dataset. (2) In [27], the training data consist of the original labeled data and the synthetic data, i.e., the labeled dataset is expanded, whereas in the proposed method, the size of the training data is the same as that of the original labeled data; i.e., the labeled dataset is not expanded but the corresponding feature space is augmented. In addition, unlike with the SMOTE method, the proposed method does not need post-processing. The reason is that the proposed method only uses the synthetic data to augment the feature space of labeled data and the final classifier is learned on the labeled dataset with the augmented features composed of the original features and the common hidden features.

3.2.2. Computation of r^* and $\boldsymbol{\Theta}^*$

As seen above, r is the dimensionality of the common hidden space and $\boldsymbol{\Theta}$ is the matrix about the common hidden space between the labeled dataset and the synthetic dataset. As seen from the formulation of Eq. (12), the labeled and synthetic datasets are projected by the linear transformation $\boldsymbol{\Theta}$, whose rows are orthonormal onto a new feature subspace in which the integrated square error between the probability distributions of the labeled and synthetic data is minimized. In other words, $\boldsymbol{\Theta}$ is essential and common to the data from the labeled and synthetic datasets. Therefore, in this study, we use the principal component analysis method on data from the labeled and synthetic datasets to extract the essential information common to them. Specifically, we can select the top r eigenvalues and the corresponding eigenvectors to form the linear transformation $\boldsymbol{\Theta}^*$.

3.2.3. Computation of $\mathbf{w}^*$ and $\mathbf{v}^*$

To solve Eq. (12) conveniently, the auxiliary variable $\boldsymbol{u} = \boldsymbol{w} + \boldsymbol{\Theta}^T\boldsymbol{v}$ is introduced. Hence, Eq. (12) can be expressed as follows:

$$\begin{aligned}
&\arg\min_{\boldsymbol{u},\boldsymbol{v},\boldsymbol{\Theta},\xi_i} J_3 = \arg\min_{\boldsymbol{u},\boldsymbol{v},\boldsymbol{\Theta},\xi_i} \frac{1}{2}\left\|\boldsymbol{u} - \boldsymbol{\Theta}^T\boldsymbol{v}\right\|^2 + \frac{\lambda_1}{2}\|\boldsymbol{u}\|^2 + C\sum_{i=1}^{n_l}\xi_i + \frac{\lambda_2}{2}\boldsymbol{v}^T\boldsymbol{\Theta}\sum_{i=1}^{n_l}\sum_{j=1}^{n_s}(\boldsymbol{x}_i^L - \boldsymbol{x}_j^S)(\boldsymbol{x}_i^L - \boldsymbol{x}_j^S)^T\boldsymbol{\Theta}^T\boldsymbol{v} \\
&s.t.\ y_i\boldsymbol{u}^T\boldsymbol{x}_i \geq 1 - \xi_i\ \xi_i \geq 0\ i = 1, 2, \ldots, n_l \\
&\boldsymbol{\Theta}\boldsymbol{\Theta}^T = \mathbf{I}_{r\times r}
\end{aligned} \tag{13}$$

Once parameters $\boldsymbol{\Theta}$ and r are computed, Eq. (13) can be expressed as follows:

$$\begin{aligned}
&\arg\min_{\boldsymbol{u},\boldsymbol{v},\xi_i} J_4 = \arg\min_{\boldsymbol{u},\boldsymbol{v},\boldsymbol{\Theta},\xi_i} \frac{1}{2}\left\|\boldsymbol{u} - \boldsymbol{\Theta}^T\boldsymbol{v}\right\|^2 + \frac{\lambda_1}{2}\|\boldsymbol{u}\|^2 + C\sum_{i=1}^{n_l}\xi_i + \frac{\lambda_2}{2}\boldsymbol{v}^T\boldsymbol{\Theta}\sum_{i=1}^{n_l}\sum_{j=1}^{n_s}(\boldsymbol{x}_i^L - \boldsymbol{x}_j^S)(\boldsymbol{x}_i^L - \boldsymbol{x}_j^S)^T\boldsymbol{\Theta}^T\boldsymbol{v} \\
&s.t.\ y_i\boldsymbol{u}^T\boldsymbol{x}_i \geq 1 - \xi_i\ \xi_i \geq 0\ i = 1, 2, \ldots, n_l
\end{aligned} \tag{14}$$

There are two parameters in Eq. (14) to be optimized. Therefore, the alternating iterative method [9,31] is adopted for optimization. Thus, two main steps are included in the following iterative procedure:

(1) Fix parameter $\boldsymbol{v}$ in Eq. (14) and then optimize $\boldsymbol{u}$;
(2) Fix parameter $\boldsymbol{u}$ in Eq. (14) and then optimize $\boldsymbol{v}$.

The above steps are executed iteratively until some conditions are satisfied. For optimizing Eq. (14), two theorems are provided.

Theorem 1. *Suppose that $\boldsymbol{v}$ is fixed; then, $\boldsymbol{u}$ can be optimized with the following update rule*:

$$\boldsymbol{u} = \frac{\sum_{i=1}^{n_l}\alpha_i y_i\boldsymbol{x}_i + \boldsymbol{\Theta}^T\boldsymbol{v}}{1 + \lambda_1} \tag{15}$$

Proof. Suppose that $\boldsymbol{v}$ is fixed; then, the objective function in Eq. (14) can be expressed as

$$\begin{aligned}
&\arg\min_{\boldsymbol{u},\xi_i} J_5 = \arg\min_{\boldsymbol{u},\xi_i} \frac{1}{2}\left\|\boldsymbol{u} - \boldsymbol{\Theta}^T\boldsymbol{v}\right\|^2 + \frac{\lambda_1}{2}\|\boldsymbol{u}\|^2 + C\sum_{i=1}^{n_l}\xi_i \\
&s.t.\ y_i\boldsymbol{u}^T\boldsymbol{x}_i \geq 1 - \xi_i\ \xi_i \geq 0\ i = 1, 2, \ldots, n_l
\end{aligned} \tag{16}$$

The duality of Eq. (16) can be expressed as follows:

$$\begin{aligned}
&\arg\min_{\alpha_i} J_6 = \arg\min_{\alpha_i} \frac{1}{2}\sum_{i=1}^{n_l}\sum_{j=1}^{n_l}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i{}^T\boldsymbol{x}_j + \sum_{i=1}^{n_l}\alpha_i(\boldsymbol{v}^T\boldsymbol{\Theta}y_i\boldsymbol{x}_i - 1 - \lambda_1) \\
&s.t.\ 0 \leq \alpha_i \leq C
\end{aligned} \tag{17}$$

With the optimal solution of Eq. (17), i.e., $\alpha_i (i = 1, 2, \ldots, n_l)$ and the duality theory, the corresponding optimal solution of $\boldsymbol{u}$ in the primal problem can be derived in the form of Eq. (15). □

Theorem 2. *Suppose that $\boldsymbol{u}$ is fixed; then, $\boldsymbol{v}$ can be optimized with the following update rule*:

$$\boldsymbol{v} = \boldsymbol{\Theta}\left(\mathbf{I}_{d\times d} - \lambda_2\sum_{i=1}^{n_l}\sum_{j=1}^{n_s}(\boldsymbol{x}_i^L - \boldsymbol{x}_j^S)(\boldsymbol{x}_i^L - \boldsymbol{x}_j^S)\right)^{-1}\boldsymbol{u} \tag{18}$$

Proof. Suppose that $\boldsymbol{u}$ is fixed; then, the objective function in Eq. (14) can be expressed as

$$\arg\min_{\boldsymbol{v}} J_7 = \arg\min_{\boldsymbol{v}} \frac{1}{2}\left\|\boldsymbol{u} - \boldsymbol{\Theta}^T\boldsymbol{v}\right\|^2 + \frac{\lambda_2}{2}\boldsymbol{v}^T\boldsymbol{\Theta}\sum_{i=1}^{n_l}\sum_{j=1}^{n_s}\left(\boldsymbol{x}_i^L - \boldsymbol{x}_j^S\right)\left(\boldsymbol{x}_i^L - \boldsymbol{x}_j^S\right)^T\boldsymbol{\Theta}^T\boldsymbol{v} \tag{19}$$

The necessary conditions of Eq. (19) can be obtained from $\frac{\partial J_7}{\partial \boldsymbol{v}} = \mathbf{0}_{r\times 1}$; hence, the solution of $\boldsymbol{v}$ in the form of Eq. (18) can be obtained.

Table 2
The proposed method OCHS-SSC.

```
Algorithm OCHS-SSC
Input: The labeled dataset $L$ and the unlabeled dataset $U$, regularization parameters $C, \lambda_1, \lambda_2$, oversampling factor $f$, number of neighbors $k$.
output: Orthonormal transformation parameter $\boldsymbol{\Theta}$, the projective vector in the original space $\boldsymbol{w}$ and the projective vector $\boldsymbol{v}$ in the common hidden feature space.
Initialization: $\boldsymbol{u}_0 \in \mathbb{R}^d$, $iter = 0$, set the maximum number $itermax$ of iterations and the threshold $\varepsilon$ of error.
$S = \mathrm{SDG}(L, U, k, f)$.
$[r, \Theta] = \mathrm{PCA}(L, S)$ // For the labeled data and the synthetic data, the principal component analysis method is adopted. Then, we can select the top $r$ eigenvalues and the corresponding eigenvectors to form the linear transformation $\boldsymbol{\Theta}^*$//.
repeat
    $\boldsymbol{v}_{iter} = \boldsymbol{\Theta}(\mathbf{I}_{d\times d} - \lambda_2 \sum_{i=1}^{n_l} \sum_{j=1}^{n_s} (\boldsymbol{x}_i^L - \boldsymbol{x}_j^S)(\boldsymbol{x}_i^L - \boldsymbol{x}_j^S))^{-1} \boldsymbol{u}_{iter}$
    $iter = iter + 1$
    $\boldsymbol{u}_{iter} = \frac{\sum_{i=1}^{n_l} \alpha_i y_i \boldsymbol{x}_i + \boldsymbol{\Theta}^T \boldsymbol{v}_{iter-1}}{1+\lambda_1}$ where $\alpha_i$ is the optimal solution of Eq. (17).
until $||\boldsymbol{u}_{iter} - \boldsymbol{u}_{iter-1}|| < \varepsilon\, or ||\boldsymbol{v}_{iter} - \boldsymbol{v}_{iter-1}|| < \varepsilon\, or(iter > itermax)$.
$\boldsymbol{w} = \boldsymbol{u}_{iter} - \boldsymbol{\Theta}^T \boldsymbol{v}_{iter}$, $\boldsymbol{v} = \boldsymbol{v}_{iter}$.
return $r, \boldsymbol{\Theta}, \boldsymbol{w}, \boldsymbol{v}$ .
```

3.3. The proposed method OCHS-SSC

Based on the above derivations and discussions, we state the proposed semi-supervised classification method OCHS-SSC through oversampling and the common hidden space in Table 2.

3.4. Convergence analysis

Because the optimization of Eq. (14) is alternatively iterative, the convergence of the proposed method OCHS-SSC essentially addresses the convergence of the optimization of Eq. (14). Discussions of the convergence of Eq. (14) are provided here:

(1) When $\boldsymbol{v}_{iter}$ is fixed, from Theorem 1 we know $\boldsymbol{u}_{iter+1}$ is the global optimal solution of $J_5(\boldsymbol{v}_{iter}, \boldsymbol{\Theta}_{iter}, \boldsymbol{u})$, i.e., $J_5(\boldsymbol{v}_{iter}, \boldsymbol{\Theta}_{iter}, \boldsymbol{u}_{iter}) \le J_5(\boldsymbol{v}_{iter}, \boldsymbol{\Theta}_{iter}, \boldsymbol{u}_{iter+1})$.

(2) When $\mathbf{u}_{iter}$ is fixed, $\boldsymbol{v}_{iter+1}$ is updated by the necessary condition of the local optimization of $J_7(\boldsymbol{v}, \boldsymbol{\Theta}, \boldsymbol{u}_{iter})$ according to Theorem 2. Thus, $\boldsymbol{v}_{iter+1}$ is a local optimal solution or an unfixed point according to the optimal theorem, which cannot be guaranteed to satisfy $J_7(\boldsymbol{v}_{iter}, \boldsymbol{\Theta}_{iter}, \boldsymbol{u}_{iter}) \le J_7(\boldsymbol{v}_{iter+1}, \boldsymbol{\Theta}_{iter}, \boldsymbol{u}_{iter})$.

4. Experimental results

4.1. Benchmark datasets and comparison algorithms

The effectiveness of the proposed method is verified on a broad range of datasets—six benchmark datasets in [12], which are often used for semi-supervised learning and can be downloaded from *http://olivier.chapelle.cc/ssl-book/benchmarks.html*, and seventeen UCI datasets, which can be downloaded from *http://sci2s.ugr.es/keel/datasets/*. All of the employed datasets are shown in Table 3. The datasets contain between 80 and 19,020 instances, and the number of their attributes ranges from 2 to 241.

We perform experiments from two views, transductive and inductive, for the benchmark datasets, each of which includes two sets with 10 labeled samples and 100 labeled samples, respectively. In the transductive setting, the training sets contain labeled and unlabeled samples, and the testing sets contain unlabeled samples. In the inductive setting, unlabeled samples are divided into two parts: one is for training and the other is for testing. In other words, unlabeled samples are initially split using a 10-fold cross-validation; then, 9-folds are used as the training samples with labeled samples and the remaining one form the testing set. For the seventeen UCI datasets, we follow the original partition strategy, i.e., each of the seventeen datasets is partitioned using a 10-fold cross-validation, and every fold is composed of three parts: training (including labeled and unlabeled samples), transductive (containing the real class of unlabeled samples), and testing (collecting the testing samples). Referring to the above strategy, in the transductive setting, the training set contains labeled and unlabeled samples, and the testing set contains unlabeled samples, whereas in the inductive setting, the testing set only contains the testing samples.

Except for the proposed method OCHS-SSC, the five comparison algorithms SVM, TSVM [15], LapSVM [1], S4VM [17] and SA-SSCCM [30] are adopted in our experiments.

4.2. Parameter settings

For the proposed method OCHS-SSC, referring to a comparatively large value often used in SVM, parameter C is determined from the parameter set $\{0.01, 0.1, 0.5, 1, 5, 10, 20, 100, 1000\}$. To keep almost the same magnitude as the first term in

Table 3
Summary of the adopted datasets.

ID	Datasets	Sizes	Dimensions
1	*G241c*	1500	241
2	*G241n*	1500	241
3	*Digit1*	1500	241
4	*Usps*	1500	241
5	*Coil-2*	1500	241
6	*BCI*	400	117
7	*Wisconsin*	683	9
8	*Twonorm*	7400	20
9	*Titanic*	2201	3
10	*Spectfheart*	267	44
11	*Spambase*	4597	57
12	*Saheart*	462	4
13	*Ring*	7400	20
14	*Phoneme*	5404	5
15	*Monk*	432	6
16	*Mammographic*	830	5
17	*Heart*	270	13
18	*Coil2000*	9822	85
19	*Bupa*	345	6
20	*Banana*	5300	2
21	*Austrilian*	690	14
22	*Appendicitis*	105	7
23	*Magic*	19020	10

Eq. (12), parameters λ_1, λ_2 are both determined from the parameter set $\{2^{-6}, 2^{-5}, \ldots, 2^5, 2^6\}$; parameter k is set as 5 and the oversampling factor f is determined from the parameter set $\{100\%, 200\%, 300\%, 400\%, 500\%\}$. According to the recommendation in [15], parameter ε is empirically set to 10^{-5}.

For SVM and TSVM, the regularization parameters are selected from the parameter set $\{0.01, 0.1, 0.5, 1, 5, 10, 20, 100, 1000\}$, and the SEG-SSC method [27] is adopted as a preprocessing step with the same parameter setting as in [27]. For other parameters in the comparison algorithms, we refer to the parameter settings in their original works. Concerning LapSVM, regularization parameters γ_A, γ_I are determined from parameter set $\{2^{-6}, 2^{-5}, \ldots, 2^5, 2^6\}$; the nearest-neighbor parameter k in LapSVM is determined from parameter set $\{1, 5, 15\}$ for datasets of size$<$400 and $\{1, 5, 15, 30, 60\}$ for datasets of size$>$400. Concerning S4VM, for a dataset of size$<$400, sampling size $N$is determined from parameter set $\{10, 20, 50, 100, 150\}$; for a dataset of size$>$400, sampling size $N$is determined from parameter set $\{100, 150, 200, 250, 300\}$, the number $T$of clusters is determined from parameter set $\{5, 10, 15, 20, 25\}$, and risk parameter λ is determined from parameter set $\{1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5\}$. Concerning SA-SSCCM, regularization parameter $\lambda$is determined from parameter set $\{0.1, 0.2, 0.5, 0.8, 0.9\}$, parameters $\lambda_1, \lambda_2, \eta$are determined from parameter set $\{0.01, 0.1, 0.5, 1, 5, 10, 20, 100, 1000\}$, and $\varepsilon$is set to 10^{-6}.

For nonlinear versions of all of the comparison methods, the Gaussian kernel function, i.e., $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{\gamma})$, is adopted and kernel width γ is determined from $\{\frac{\sigma^2}{32}, \frac{\sigma^2}{16}, \frac{\sigma^2}{8}, \frac{\sigma^2}{4}, \frac{\sigma^2}{2}, \sigma^2, 2\sigma^2, 4\sigma^2, 8\sigma^2, 16\sigma^2\}$, where σ is the standard deviation of the corresponding dataset.

For all comparison algorithms, a 10-fold cross-validation strategy is used to determine the final classification accuracy.

Table 4
Accuracy (mean ± std) of six comparison algorithms on benchmark datasets (transductive setting).

No. of labeled samples	ID	SVM (linear/Gaussian)	TSVM (linear/Gaussian)	LapSVM (linear/Gaussian)	S4VM (linear/Gaussian)	SA-SSCCM (linear/Gaussian)	OCHS-SSC (linear/Gaussian)
10	1	57.1±2.1/56.9±0.5	59.1±2.7/58.5±3.6	61.1±5.2/55.8±4.1	64.5±4.6/56.6±5.3	64.3±4.0/57.6±3.1	**66.9±2.4/65.4±1.4**
	2	54.9±1.9/55.1±1.3	55.1±3.1/52.9±2.1	54.6±6.4/51.8±2.6	55.1±5.3/53.3±5.2	56.7±4.4/53.8±3.6	**80.4±4.0/83.0±6.8**
	3	70.1±0.6/69.4±4.5	**79.1±4.2/81.3±7.2**	68.3±8.1/77.4±6.8	72.4±6.3/81.4±5.9	69.0±5.7/75.7±4.1	79.9±6.2/78.5±0.2
	4	67.9±0.1/71.6±5.0	71.4±1.9/73.5±3.1	78.3±2.3/75.8±1.3	77.6±3.0/75.8±3.6	81.0±0.9/80.2±2.7	**81.7±2.9/83.2±1.9**
	5	56.9±1.2/60.4±1.5	58.9±2.6/61.0±4.2	58.9±5.8/60.1±6.4	**67.0±5.2/69.3±5.0**	62.7±4.6/63.8±5.9	65.4±0.5/68.3±2.4
	6	54.6±5.0/56.0±3.2	52.8±5.6/51.3±7.8	51.9±2.9/51.7±2.2	51.6±2.7/51.1±2.4	53.8±2.8/53.3±2.8	**57.2±1.6/59.4±4.5**
100	1	71.5±5.3/69.8±3.4	**81.3±0.1/73.5±4.5**	72.9±3.2/68.8±2.7	71.6±2.4/71.7±2.5	77.6±1.8/72.6±2.9	76.6±1.1/79.5±3.2
	2	71.8±3.4/69.2±3.6	76.4±5.0/73.4±4.5	75.6±3.1/70.7±5.1	76.1±2.9/70.4±3.3	74.2±3.1/70.6±3.0	**82.4±4.1/88.1±2.1**
	3	73.5±4.8/85.2±3.9	**86.1±4.0/91.6±1.2**	74.5±3.7/91.6±1.9	73.1±2.9/90.1±2.6	81.3±1.8/93.1±2.1	80.3±3.9/81.7±2.6
	4	83.5±2.6/84.7±3.1	81.5±4.9/85.0±7.8	80.9±2.5/83.0±2.1	82.4±1.9/87.3±1.3	81.0±2.2/82.7±3.2	**88.3±12/89.5±2.7**
	5	66.1±7.3/75.2±1.6	69.6±3.9/72.7±3.2	68.6±3.5/75.7±6.5	69.5±4.6/74.6±4.8	69.0±4.0/75.1±4.4	**79.8±0.5/78.4±2.5**
	6	63.4±6.8/64.1±0.9	60.4±4.0/62.5±3.2	61.7±4.2/62.2±3.9	61.1±3.0/61.2±2.8	62.6±4.5/62.7±3.4	**75.9±7.1/78.4±1.5**

Table 5
Accuracy (mean ± std) of six comparison algorithms on benchmark datasets (inductive setting).

No. of labeled samples	ID	SVM (linear/Gaussian)	TSVM (linear/Gaussian)	LapSVM (linear/Gaussian)	S4VM (linear/Gaussian)	SA-SSCCM (linear/Gaussian)	OCHS-SSC (linear/Gaussian)
10	1	52.9±1.2/55.2±4.0	58.5±3.1/59.6±5.6	59.4±2.6/54.9±1.8	65.9±2.9/57.5±2.2	65.7±2.3/60.2±2.4	**67.8±0.4/66.1±1.5**
	2	58.5±3.2/57.1±8.7	57.2±9.1/56.0±2.5	53.2±3.0/52.5±1.8	56.8±2.9/54.7±1.5	58.9±2.2/55.3±2.7	**77.5±0.6/79.2±1.2**
	3	67.3±0.5/68.9±8.0	**75.5±3.3**/79.4±5.9	70.1±1.7/75.5±3.0	71.3±0.7/**80.6±0.4**	68.3±1.2/74.7±1.5	78.5±0.4/79.3±0.2
	4	68.2±3.0/73.0±4.3	71.0±3.6/73.8±4.0	76.9±1.8/74.2±2.5	79.1±0.9/77.5±0.3	**80.3±1.4/81.8±1.6**	80.2±0.4/84.1±0.3
	5	59.4±6.9/58.3±5.2	58.6±2.4/57.9±1.7	58.1±3.5/61.8±2.2	**69.0±0.4**/65.9±1.6	63.5±0.2/64.8±1.5	66.2±1.7/**69.4±2.7**
	6	50.9±4.3/53.7±4.4	54.0±3.9/54.9±3.2	52.5±3.3/52.7±3.0	53.5±0.5/54.9±1.2	55.8±1.2/57.2±1.7	**58.1±1.8/60.4±3.1**
100	1	71.2±3.6/72.9±3.4	**78.0±1.6**/77.9±2.4	71.1±1.2/66.9±1.3	72.5±0.4/74.0±0.2	75.7±3.4/73.8±1.3	78.2±1.6/**77.4±2.0**
	2	73.1±2.7/68.5±6.8	77.9±5.6/74.0±6.6	74.7±1.5/69.4±1.5	75.3±0.4/73.0±0.1	75.2±3.7/74.6±1.2	**81.5±1.9/84.3±2.1**
	3	75.0±4.8/85.7±1.6	**85.2±3.1/90.0±2.9**	78.4±0.2/85.7±1.6	72.6±1.7/87.4±2.5	82.7±2.0/89.1±2.6	82.5±2.3/83.8±1.0
	4	82.0±0.2/86.4±0.5	79.6±4.5/81.7±2.7	78.9±1.0/82.0±1.3	79.4±1.6/84.3±3.0	81.4±1.7/83.5±3.2	**87.1±1.4/86.5±1.8**
	5	71.2±1.0/76.7±0.9	68.9±2.1/74.0±2.3	69.4±3.4/74.1±1.2	68.4±1.7/72.6±2.1	67.2±2.5/73.7±0.8	**76.8±0.4/77.4±1.3**
	6	64.0±1.6/64.3±7.3	56.9±2.5/62.0±2.3	62.5±3.6/61.9±1.2	59.9±2.2/62.5±2.0	61.8±2.5/63.9±0.9	**76.3±0.4/76.2±1.4**

Table 6
Accuracy (mean ± std) of six comparison algorithms on seventeen UCI datasets (transductive setting).

ID	SVM (linear/Gaussian)	TSVM (linear/Gaussian)	LapSVM (linear/Gaussian)	S4VM (linear/Gaussian)	SA-SSCCM (linear/Gaussian)	OCHS-SSC (linear/Gaussian)
7	76.3±4.7/78.0±5.3	78.5±8.7/83.6±6.6	81.7±9.5/82.3±3.8	86.5±8.7/86.6±0.4	**86.7±0.6**/87.0±3.8	85.9±3.2/**88.0±0.2**
8	73.9±2.2/72.5±3.1	70.1±2.4/74.3±2.6	74.5±2.4/72.9±3.1	75.5±1.6/75.9±3.1	77.9±3.1/76.1±8.0	**80.0±1.2/78.6±3.2**
9	81.2±1.8/76.9±2.1	81.2±3.1/79.6±1.9	80.2±3.5/82.0±3.9	82.0±1.2/82.9±2.1	**84.5±1.7**/85.5±3.9	83.9±4.1/**85.6±3.7**
10	68.0±6.6/74.0±5.6	70.1±1.7/75.3±2.6	72.8±4.2/75.0±2.1	74.8±1.8/78.1±8.0	80.1±2.1/81.2±2.9	**81.0±3.7/82.3±7.6**
11	78.2±2.5/80.6±2.5	84.0±2.9/82.4±5.0	81.2±1.5/81.7±4.9	82.0±3.7/83.4±4.5	81.9±3.6/83.0±3.5	**82.8±6.8/84.6±0.1**
12	79.9±3.3/80.6±4.6	75.8±4.8/78.3±4.6	79.3±1.6/79.6±1.3	80.7±3.5/81.3±4.5	82.4±1.3/83.2±3.2	**83.9±0.5/84.4±0.5**
13	69.5±5.0/75.8±3.2	74.1±4.5/75.0±3.4	73.5±2.1/74.9±3.6	78.4±2.4/77.9±3.9	80.0±2.5/81.5±2.4	**81.3±2.9/83.0±8.9**
14	73.9±2.4/75.7±2.5	69.8±2.9/70.2±2.2	72.1±5.1/72.6±4.8	75.3±4.8/76.5±3.1	78.6±3.8/79.2±3.6	**80.0±0.1/79.8±5.8**
15	70.8±2.3/73.2±1.3	71.6±8.1/71.4±6.8	70.9±0.1/71.2±6.8	74.0±8.9/73.5±6.5	73.6±4.5/74.8±2.9	**76.5±0.1/77.6±3.9**
16	79.0±2.7/80.1±0.9	78.5±5.7/78.1±4.1	78.0±3.8/80.0±8.9	81.4±6.4/79.6±3.6	80.5±3.2/81.6±3.8	**82.7±0.6/82.9±0.1**
17	64.8±1.1/65.6±3.2	64.3±4.6/67.5±5.9	70.0±5.9/68.6±4.9	73.6±4.7/72.1±5.2	75.0±3.2/76.4±2.1	**77.6±0.6/76.9±4.6**
18	75.9±1.5/76.4±0.5	74.0±2.2/75.8±3.2	75.8±3.7/76.0±4.3	74.9±1.0/75.5±4.9	76.0±3.2/78.3±2.0	**79.0±3.1/78.9±0.5**
19	73.8±8.1/77.2±6.8	72.5±2.6/74.3±1.3	76.9±1.2/78.0±2.9	79.4±2.3/80.1±4.2	82.7±0.4/83.0±3.0	**84.0±1.2/83.5±2.7**
20	78.4±2.4/79.7±2.8	83.0±1.8/82.5±3.0	80.5±3.8/81.2±4.2	81.6±3.2/82.0±3.9	84.0±3.7/82.1±0.4	**85.9±2.0/87.0±2.7**
21	75.8±2.7/74.2±1.5	76.8±1.9/75.1±2.1	76.9±3.9/75.4±5.2	77.8±4.1/76.5±0.2	78.4±0.4/80.3±4.8	**81.2±3.5/80.2±2.4**
22	71.0±3.5/73.5±6.5	72.9±1.2/76.8±2.7	73.6±2.1/75.0±12	72.9±3.9/76.3±3.1	76.9±4.1/78.2±3.1	**78.4±0.3/79.3±2.1**
23	77.2±2.5/75.8±7.4	76.7±0.9/75.9±0.3	76.4±6.4/77.0±3.9	77.2±3.5/78.4±2.8	76.9±4.1/79.3±0.1	**80.1±4.9/83.0±3.2**

Table 7
Accuracy (mean ± std) of six comparison algorithms on seventeen UCI datasets (inductive setting).

ID	SVM (linear/Gaussian)	TSVM (linear/Gaussian)	LapSVM (linear/Gaussian)	S4VM (linear/Gaussian)	SA-SSCCM (linear/Gaussian)	OCHS-SSC (linear/Gaussian)
7	74.0±1.5/75.3±2.7	78.0±2.3/80.0±7.6	80.9±7.4/81.9±2.5	86.7±6.8/87.1±0.9	87.2±0.4/87.9±2.1	**87.6±6.9/89.1±0.9**
8	72.0±2.8/71.8±2.5	68.1±2.1/72.5±9.0	75.5±1.1/72.1±1.2	74.9±4.9/74.2±1.2	78.5±1.0/75.8±3.9	**79.8±2.1/77.2±2.8**
9	76.1±1.2/79.8±1.6	79.1±4.9/78.5±5.1	79.9±2.1/82.1±1.8	81.8±5.4/83.0±2.7	84.7±1.4/83.5±2.3	**84.9±3.7/84.2±2.1**
10	67.6±7.4/73.5±8.5	73.0±4.1/75.2±3.5	71.7±1.5/75.1±1.8	75.1±2.1/77.9±3.9	79.3±3.4/80.9±4.0	**81.2±2.5/81.8±4.2**
11	77.4±1.9/79.3±2.0	80.3±1.9/80.1±2.8	81.0±4.2/80.9±7.6	81.9±3.0/83.7±8.0	82.7±3.1/83.5±3.7	81.9±9.0/**85.3±1.2**
12	79.0±2.0/80.5±1.4	76.7±0.3/81.1±1.7	80.0±1.0/78.9±4.0	81.9±2.1/82.8±2.9	83.5±2.1/84.1±5.0	82.6±4.5/83.9±1.7
13	73.5±1.6/74.3±2.0	73.0±2.9/75.6±3.5	74.5±9.0/75.1±2.8	78.9±1.9/76.2±5.0	80.7±3.8/82.8±6.1	**82.7±2.1/83.9±2.3**
14	71.8±3.5/72.9±1.8	70.2±2.1/73.5±3.1	70.9±6.8/73.2±1.2	76.1±3.2/77.9±2.3	77.9±5.1/80.5±1.9	**81.2±8.0**/80.0±4.0
15	74.0±2.1/70.9±3.7	73.0±2.0/73.2±0.9	71.0±4.5/72.9±6.2	74.0±1.5/74.2±3.4	72.8±2.9/75.1±2.8	**74.3±0.7/76.8±2.1**
16	76.4±3.5/75.9±2.1	77.9±0.5/79.2±1.3	77.6±3.3/78.8±0.8	81.0±3.2/80.1±6.1	79.8±4.0/80.2±2.4	**82.0±0.1/83.6±0.6**
17	66.1±4.1/67.0±3.9	62.8±2.6/65.9±0.7	69.1±3.2/67.2±2.9	74.2±2.9/73.5±3.0	73.8±4.1/75.9±2.7	**76.8±1.2/77.1±3.5**
18	75.1±1.9/72.9±2.7	73.7±2.5/75.0±2.1	73.2±4.8/74.8±3.5	75.9±3.9/76.8±3.6	75.0±3.9/76.9±3.1	**78.5±2.3/79.0±2.1**
19	75.9±3.5/76.1±2.6	71.8±3.6/75.7±3.3	75.3±6.2/76.8±1.9	79.7±9.0/79.9±2.9	80.6±8.1/82.1±3.0	**82.8±3.0/84.1±2.5**
20	81.4±1.0/77.9±3.8	82.1±2.6/78.9±3.1	79.5±8.0/80.3±3.9	83.2±4.1/83.6±4.1	82.8±6.5/81.9±0.9	**84.2±1.8/85.9±0.9**
21	74.2±0.7/73.3±5.0	77.3±8.5/79.2±7.4	75.2±2.7/76.3±3.0	78.1±7.6/77.2±0.8	78.8±1.7/79.9±1.8	**80.9±1.3/81.4±1.2**
22	73.1±3.9/75.0±8.0	73.5±0.2/75.8±1.1	74.9±4.2/73.8±3.0	71.8±4.9/74.5±2.8	76.1±3.9/77.5±2.9	**79.2±0.9/79.5±1.6**
23	74.2±0.9/69.8±1.8	76.4±1.4/75.9±1.3	76.7±7.2/76.1±2.8	76.2±4.3/76.5±3.9	77.0±2.5/80.1±4.5	**79.0±2.6/82.7±2.8**

4.3. Comparisons of classification accuracy

We perform extensive experiments with the datasets demonstrated in sub-section IV-A and the parameter settings illustrated in sub-section IV-B. The obtained results are reported in Tables 4–7.

From Tables 4–7, we can easily make the following observations:

Table 8
Comparison of Friedman test and Holm's post hoc test on all of the algorithms on all datasets (transductive setting).

Algorithms	Average accuracy		
	F-Rank	>=	<
SVM (linear)	9.4138	4	7
SVM(Gaussian)	8.2414	3	8
TSVM(linear)	8.4138	3	8
TSVM(Gaussian)	7.7931	3	8
LapSVM(linear)	9.0000	2	9
LapSVM(Gaussian)	8.0690	4	7
S4VM(linear)	6.9310	6	5
S4VM(Gaussian)	6.3103	7	4
SA-SSCCM(linear)	5.2414	7	4
SA-SSCCM (Gaussian)	4.3448	8	3
OCHS-SSC(linear)	2.5172	11	0
OCHS-SSC (Gaussian)	1.7241	11	0

(1) The overall performance of SVM is worse than that of the other comparison algorithms because SVM is in essence a supervised method; it only considers a few labeled samples to learn the final classifier.
(2) Both TSVM and LapSVM are semi-supervised classification methods. Their performances are better than SVM in most cases and are worse than SVM in a few cases. This observation reflects the fact that the usage of unlabeled samples is occasionally dangerous if unlabeled samples are not used in an appropriate manner.
(3) The overall performances of both S4VM and SA-SSCCM are comparable and even better than the other semi-supervised classification algorithms because they leverage unlabeled samples to learn the final classifier with a relatively safe manner through a certain strategy.
(4) In most cases, the proposed method OCHS-SSC is better than the other semi-supervised classification algorithms. The reason is that the method does not use unlabeled samples directly; it learns the final classifier in an extended feature space that consists of the original feature space and the common hidden feature space between labeled samples and the synthetic samples.

Let us observe the sensitivity of a specified parameter in the proposed method OCHS-SSC. To save space in the paper, we only report here the experimental results concerning the parameter sensitivity of OCHS-SSC on seven datasets rather than all datasets. These seven datasets, *G241C*, *BCI*, *Twonorm*, *Heart*, *Titanic*, *Coil2000* and *Magic*, are representative of all of the adopted 23 datasets in the senses of both the number of samples and the number of dimensions. We initially fix all other parameters with their optimal values and then compare the performance under different values of this specified parameter. Fig. 2 illustrates the experimental results on parameter sensitivity for the inductive setting with a linear kernel. Please note that we do not provide additional experimental results here because the experimental results for the inductive setting with Gaussian kernel and transductive setting with both linear and Gaussian kernels are the same as in Fig. 2. In Fig. 2, for datasets *G241c* and *BCI*, the number of labeled samples is ten.

Fig. 2 shows that the performance of the proposed method OCHS-SSC is quite insensitive to the setting of all of the parameters. This lack of sensitivity is an obvious advantage when the proposed method is applied to practical scenarios.

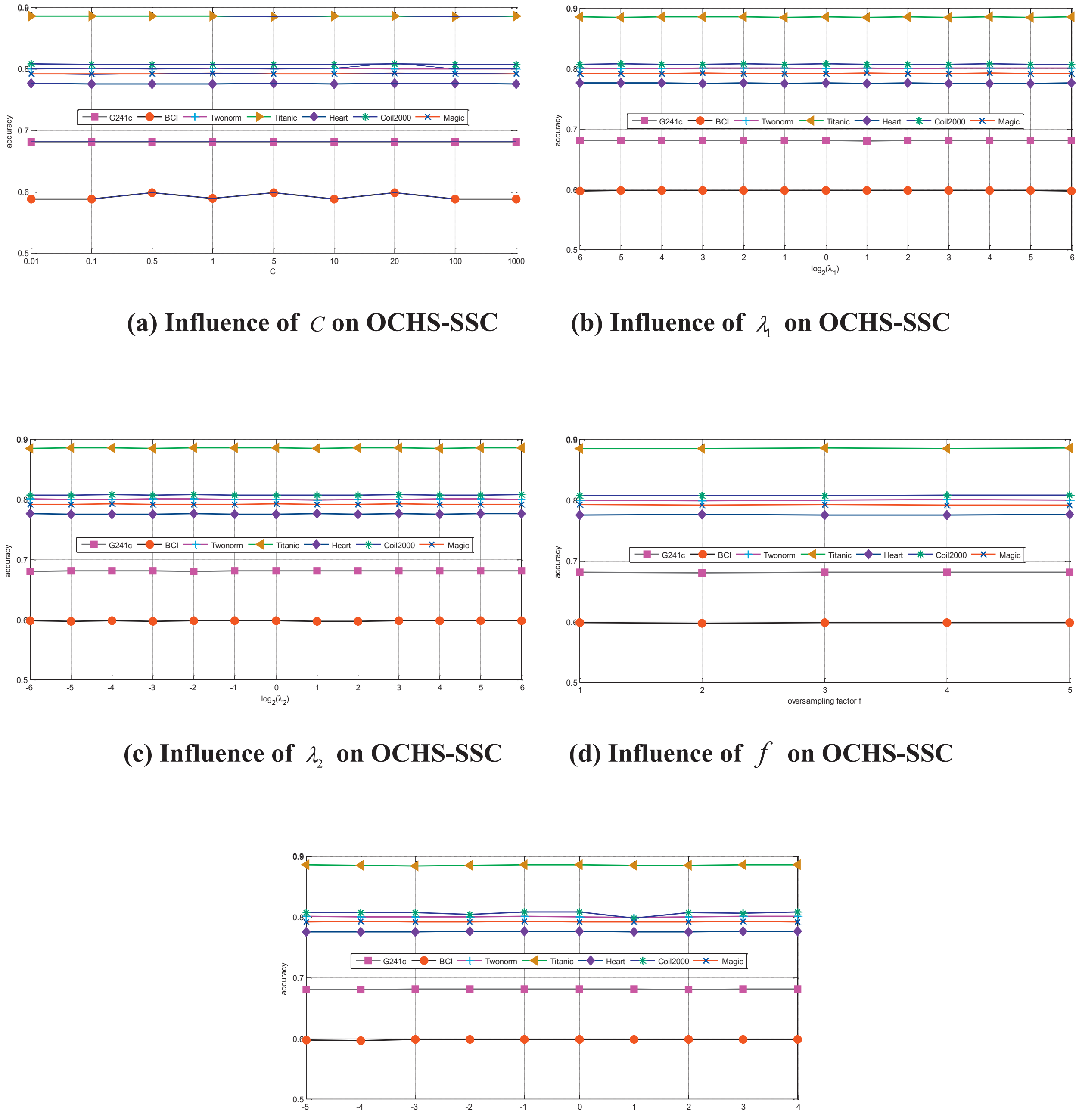
4.4. Comparison of running time

Let us observe the computational efficiency of the proposed method OCHS-SSC. Similarly, to save space in the paper, we only report the running time of the proposed methods OCHS-SSC, S4VM and SA-SSCCM on two benchmark datasets and five UCI datasets because the three comparison algorithms are the best among all comparison algorithms in the sense of classification accuracy. Moreover, the seven datasets are representative of all six of the benchmark datasets and the seventeen UCI datasets. Fig. 3 illustrates the running time of S4VM, SA-SSCCM and OCHS-SSC with linear and Gaussian kernels. In terms of Fig. 3, we can readily see that OCHS-SSC is comparable to the comparison algorithms in the sense of the magnitude of running time and even has a slightly lower computational burden than do the comparison algorithms in most cases. We can also see that the running time increases with the increasing number of samples. Therefore, how to speed up the proposed method for large data is an open problem we should investigate in the near future.

4.5. Statistical comparison

To examine the significant differences between all of the comparison algorithms, we perform the Friedman test [8] with the significance level $\alpha = 0.05$ and the post hoc Holm's test [13] for multiple comparisons between multiple methods with the same value of α.

Referring to Tables 4–7, Tables 8 and 9 record the statistical results for transductive and inductive settings, respectively. In these two tables, the column "F-Rank" shows the rankings computed by the Friedman test, which are the average values



(a) Influence of C on OCHS-SSC

(b) Influence of λ_1 on OCHS-SSC

(c) Influence of λ_2 on OCHS-SSC

(d) Influence of f on OCHS-SSC

(e) Influence of γ on OCHS-SSC

Fig. 2. Parameter influence with a linear kernel on five representative datasets.

of the rankings achieved by each algorithm on all datasets. Column “>=” shows the number of algorithms which are worse than or equal to the algorithm in the row according to the Holm’s test; column “<” shows the number of algorithms that are better than the algorithm in the row according to the Holm’s test.

According to the F-Rank in Tables 8 and 9, the Friedman test reveals that there exist significant differences in the accuracy index. In addition, the Holm’s post hoc test shows that the proposed method OCHS-SSC outperforms other comparison algorithms adopted in our experiments. These results also reveal that the usage of the adopted oversampling technique and the concept of the common hidden space does in fact enhance the generalization capability of semi-supervised classification learning.

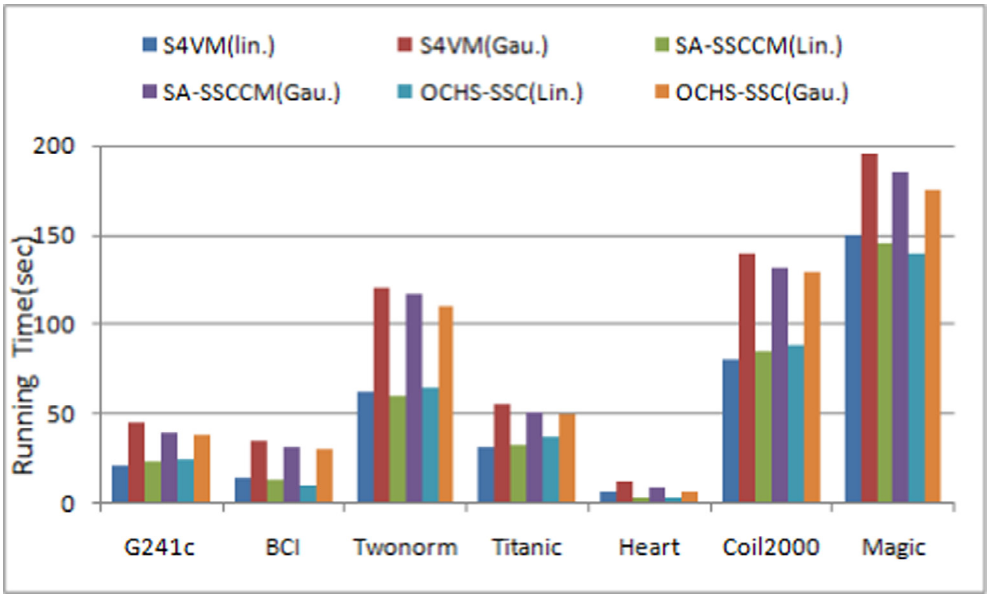


Fig. 3. Running time of OCHS-SSC, S4VM and SA-SSCCM on seven datasets.

Table 9
Comparison of Friedman test and Holm's post hoc test on all of the algorithms on all datasets (inductive setting).

Algorithms	Average accuracy		
	F-Rank	>=	<
SVM (linear)	9.6207	2	9
SVM(Gaussian)	8.9655	4	7
TSVM(linear)	8.8276	3	8
TSVM(Gaussian)	7.3103	4	7
LapSVM(linear)	8.9655	2	9
LapSVM(Gaussian)	8.3793	2	9
S4VM(linear)	6.7241	7	4
S4VM(Gaussian)	5.5172	8	3
SA-SSCCM(linear)	5.4828	7	4
SA-SSCCM (Gaussian)	3.9310	8	3
OCHS-SSC(linear)	2.5517	10	1
OCHS-SSC (Gaussian)	1.7241	10	1

5. Conclusions and future work

In this study, a new semi-supervised classification method is proposed. The method initially expands the number of labeled samples through oversampling technology and generates a certain number of synthetic samples; then, it learns the common hidden space between labeled samples and the synthetic samples. Finally, it learns the final classifier in the original space and the common hidden feature space. The proposed method has both theoretical and experimental advantages over other semi-supervised classification methods.

Although the proposed method has demonstrated promising results, open issues remain. For example, how to appropriately determine the dimensionality of the common feature space and how to scale up the proposed method for large datasets are subjects worthy of study in the near future.

Acknowledgments

This work was supported in part by Hong Kong Polytechnic University under Grant G-UA3W, by the National Natural Science Foundation of China under Grants 61272210, 61572236, by the Fundamental Research Funds for the Central Universities under Grant JUSRP51321B, by the Jiangsu Province Outstanding Youth Fund under Grant BK20140001, by the Natural Science Foundation of Jiangsu Province under Grant BK20130155, and by the Shandong Province Higher Educational Science and Technology Program under Grant J14LN05.

References

[1] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework from learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (8) (2006) 2399–2434.

[2] S. Ben-David, T. Lu, D. Pal, Does unlabeled data provably help? Worst-case analysis of the sample complexity of semi-supervised learning, in: Proc. COLT2008, 2008, pp. 33–44.

[3] A. Blum, S. Chawla, Learning from labeled and unlabeled data using graph mincuts, in: Proc. ICML2001, 2001, pp. 19–26.

[4] Y. Cao, H. He, H. Huang, Lift: a new framework of learning from testing data for face recognition, Neurocomputing 74 (2011) 916–929.

[5] O. Chapelle, B. Schlkopf, A. Zien, Semi-Supervised Learning, 1st ed., MIT Press, Cambridge, MA, USA, 2006.

[6] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J, Artif. Intell. Res. 16 (1) (2002) 321–357.

[7] K. Chen, S. Wang, Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions, IEEE Trans.Pattern Anal. Mach. Intell. 33 (1) (2011) 129–143.

[8] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[9] Z.H. Deng, K.S. Choi, F.L. Chung, S.T. Wang, Enhanced soft subspace clustering integrating within-cluster and between-cluster information, Patt. Recog. 43 (3) (2010) 767–781.

[10] Z.H. Deng, F.L. Chuang, S.T. Wang, FRSDE: fast reduced set density estimator using minimal enclosing ball approximation, Patt. Recog. 41 (2008) 1363–1372.

[11] Z.H. Deng, Y.Z. Jiang, F.L. Chung, H. Ishibuchi, S.T. Wang, Knowledge-Leverage based fuzzy system and its modeling, IEEE Trans, Fuzzy Syst. 21 (4) (2013) 597–609.

[12] H. Gan, N. Sang, R. Huang, Self-training-based face recognition using semi-supervised linear discriminant analysis and affinity propagation, J. Opt. Soc. America A 31 (2014) 1–6.

[13] S. García, F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, J. Mach. Learn. Res. 9 (2008) 2677–2694.

[14] H. Han, W. Wang, B. Mao, Borderline-SMOTE: a new oversampling method in imbalanced data sets learning, in: Proceedings of the International Conference on Intelligent Computing, Hefei,China, 2005, pp. 878–887.

[15] T. Joachims, Transductive inference for text classification using support vector machines, in: Proc. 16th Int. Conf. Mach. Learn., Morgan Kaufmann Publishers, San Francisco, 1999, pp. 200–209.

[16] M. Li and Z.H. Zhou, SETRED: self-training with editing. in: Lecture Notes in Computer Science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), 3518: 611–621, 2005.

[17] Y.F. Li, Z.H Zhou, Towards making unlabeled data never hurt, in: Proc. 28th Int. Conf. Mach. Learn., 2011, pp. 1081–1088.

[18] X.L. Liu, S.H. Pan, Z.F. Hao, Z.Y. Lin, Graph-based semi-supervised learning by mixed label propagation with a soft constraint, Inform. Sci. 277 (1) (2014) 327–337.

[19] K. Lu, Q. Wang, J. Xue, W.G. Pan, 3D model retrieval and classification by semi-supervised learning with content-based similarity, Inf. Sci. 281 (10) (2014) 703–713.

[20] D.J. Miller, H.S. Uyar, A mixture of experts classifier with learning based on both labeled and unlabelled data, in: Proc. NIPS1997, 9, 1997, pp. 571–577.

[21] K. Nigam, A.K. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using EM, Mach. Learn 39 (2) (2000) 103–134.

[22] Z. Qi, Y. Xu, L. Wang, Y. Song, Online multiple instance boosting for object detection, Neurocomputing 74 (2011) 1769–1775.

[23] M. Roy, S. Ghosh, A. Ghosh, A novel approach for change detection of remotely sensed images using semi-supervised multiple classifier system, Inform. Sci. 269 (10) (2014) 35–47.

[24] F. Schwenker, E. Trentin, Pattern classification and clustering: a review of partially supervised learning approaches, Patt. Recog. Lett 37 (2014) 4–14.

[25] A. Singh, R. Nowak, X. Zhu, Unlabeled data: now it helps, now it doesn't, in: Proc. NIPS2009, 21, 2009, pp. 1513–1520.

[26] I. Triguero, S. García, F. Herrera, Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study, Knowl. Inform. Syst. 42 (2015) 245–284.

[27] I. Triguero, S. García, F. Herrera, SEG-SSC: a framework based on synthetic examples generation for self-labeled semi-supervised classification, IEEE Trans. Cybernetics 45 (4) (2015) 622–634.

[28] S. Vaerenbergh, I. Santamaria, P. Barbano, Semi-supervised handwritten digit recognition using very few labeled data, in: Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing, 2011, pp. 2136–2139.

[29] B. Varadarajan, D. Yu, L. Deng, A. Acero, Using collective information in semi-supervised learning for speech recognition, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2009, pp. 4633–4636.

[30] Y.Y. Wang, S.C. Chen, Safety-aware semi-supervised classification, IEEE Trans. Neural Netw. Learn. Syst. 24 (11) (November 2013) 1763–1772.

[31] S. Yang, S. Yan, C. Zhang, X.O. Tang, Bilinear analysis for kernel selection and nonlinear feature extraction, IEEE Trans. Neural Netw. 18 (5) (2007) 1442–1452.

[32] D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods, in: Proceedings of the 33rd annual meeting of the association for computational linguistics, 1995, pp. 189–196.

[33] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Scholkopf, Learning with local and global consistency, in: Proc. NIPS2004, 16, 2004, pp. 595–602.

[34] Z.H. Zhou, M. Li, Tri-training: exploiting unlabeled data using three classifiers, 17, 2005, pp. 1529–1541.

[35] Z.H. Zhou, M. Li, Semi-supervised learning by disagreement, Knowledge and Information Systems 24 (3) (2010) 415–439.

[36] X. Zhu, Semi-supervised learning literature survey, Ph.D. dissertation, Dept. Comput. Sci., Wisconsin-Madison, Univ., Madison, WI, USA, Jul. 2008.

[37] X. Zhu, Z. Ghahramani, J.D. Lafferty, Semi-supervised learning using Gaussian fields and harmonic functions, in: Proc. ICML2003, 2003, pp. 912–919.

[38] X. Zhu, A.B. Goldberg, Introduction to Semi-Supervised Learning, 1st edition, Morgan and Claypool, 2009.

[39] X.Q. Zhu, X.D. Wu, Class noise vs. attribute noise: a quantitative study of their impacts, Artif. Intell. Rev. 22 (3) (2004) 177–210.