Neural Computing and Applications (2019) 31:6559–6574
https://doi.org/10.1007/s00521-018-3478-1

ORIGINAL ARTICLE

Structure regularized self-paced learning for robust semi-supervised pattern classification

Nannan Gu[1] · Pengying Fan[2] · Mingyu Fan[3] · Di Wang[3]

Received: 2 November 2017 / Accepted: 3 April 2018 / Published online: 19 April 2018

Abstract
Semi-supervised classification is a hot topic in pattern recognition and machine learning. However, in presence of heavy noise and outliers, the unlabeled training data could be very challenging or even misleading for the semi-supervised classifier. In this paper, we propose a novel structure regularized self-paced learning method for semi-supervised classification problems, which can efficiently learn partially labeled training data sequentially from the simple to the complex ones. The proposed formulation consists of three components: a cost function defined by a mixture of losses, a functional complexity regularizer, and a self-paced regularizer; and the corresponding optimization algorithm involves three iterative steps: classifier updating, sample importance calculating, and pseudo-labeling. In the proposed method, the cost function for classifier updating and sample importance calculating is defined as a combination of the label fitting loss and manifold smoothness loss. Then, the importance of the pseudo-labeled and unlabeled samples is adaptively calculated by the novel cost. Unlabeled samples with high importance values are pseudo-labeled with their current predictions. In this way, labels are efficiently propagated from the labeled samples to the unlabeled ones in the robust self-paced manner. Experimental results on several benchmark data sets are provided to show the effectiveness of the proposed method.

Keywords Semi-supervised classification · Pattern classification · Self-paced learning · Manifold learning · Locally linear coding

1 Introduction

Recently, semi-supervised classification (SSC) has received considerable interest in pattern recognition and machine learning. It can utilize a large amount of unlabeled data to help the labeled data build a better classifier. This is found to be very useful in many real-world applications where labeled samples are expensive to obtain and unlabeled data are cheap. Successful applications of SSC include image classification [1], text analysis [2], and bioinformatics [3]. So far, many SSC methods have been proposed and studied, such as the generative-based method [4, 5], self-training [6, 7], co-training [8, 9], transductive support vector machines [10], sparse-based models [11],and graph-based methods [12–17].

Among various kinds of SSC approaches, the graph-based SSC (GSSC) methods have attracted much attention due to their success in applications and the computational efficiency. GSSC methods generally need to define a graph $\mathcal{G} = (V, E)$ over the training data, where the set V consists of both the labeled and unlabeled samples, and E denotes the set of edges. There is a weight matrix $\mathbf{W}$ for graph $\mathcal{G}$ whose entries represent the similarities between pair-wise samples. Then, based on certain assumptions, the label information of labeled samples is propagated to unlabeled

✉ Mingyu Fan
fanmingyu@amss.ac.cn

Nannan Gu
gu_nannan@126.com

Pengying Fan
fpy1230@163.com

Di Wang
wangdi@wzu.edu.cn

[1] School of Statistics, Capital University of Economics and Business, Beijing 100070, China

[2] School of Economics, Beijing Technology and Business University, Beijing 100048, China

[3] College of Mathematics and Information Science, Wenzhou University, Wenzhou 325035, China

samples across the graph. Important assumption for SSC is the cluster assumption [18–20],which assumes nearby samples are likely to belong to the same class and points on the same cluster are likely to belong to the same class as well. Typically, Zhu et al. proposed an Gaussian random field (GRF) method [12, 13], in which the learning problem is formulated as a Gaussian random field on data graph, and then harmonic functions are employed to propagate the label information. Zhou et al. proposed an algorithm called learning with local and global consistency (LGC) [14], in which an iterative framework is constructed on the graph over data manifold. Belkin et al. proposed the manifold regularization (MR) framework [15] which considers both the complexity of the classifier in ambient space and the smoothness of the classifier on data manifold.

Despite the success of SSC [21], there are still some open problems that have not been addressed thoroughly. One important problem is how to efficiently deal with data that are with various level of noise and even outliers. The value of the discriminative information in the training data varies drastically from one sample to another. Many SSC methods simply treat all samples equally, regardless of the different values/contributions of the data samples on classifier training, and thus maybe suboptimal in the view of classification. To address the problem, in this paper, we propose a novel structure regularized spaced learning (SSPL) method for robust semi-supervised classification. The proposed method is based on the MR [15] framework that consists of three terms: a fitting term for the labeled points, a regularization term that controls the complexity of the classifier in the ambient space, and another regularization term that controls the smoothness of the classifier with respect to the intrinsic distribution of data. Furthermore, the proposed method makes use of the strategy of a recently proposed learning regime, Self-Paced Learning (SPL), to evaluate sample importance according to the sample-wise cost and then assign pseudo-labels to important unlabeled samples. SPL [22] is motivated by the learning principle of human/animal that trains a rough model on easy/important samples first, and then automatically incorporates more complex/(less important) samples in the self-paced fashion. Theoretical analysis of the robustness of SPL in the presence of extreme outliers or heavy noises has been provided by Meng et al. [23]. Because of its generality, the SPL theory has been applied to various tasks, such as Multimedia Event Detection (MED) [24], co-saliency detection [25], face identification [26], object tracking [27], and specific-class segmentation learning [28]. Especially, the SPL regime has been integrated into the system developed by CMU Informedia team, and achieved the leading performance in the challenging TRECVID MED/MER competition organized by NIST in 2014 [29].

Referring to the self-paced theory, the proposed SSPL method iterates among three key steps: classifier training, sample importance calculating, and pseudo-labeling. To train the classifier, we utilize the locally linear reconstruction to control the smoothness of the classification function with respect to data manifold distribution, and we also consider minimization of the label predicting error and the complexity of the classification function in the reproducing kernel Hilbert space of the classifier. To define the sample importance, we propose a novel cost function which consists of a mixture of losses. The new cost function combines the label fitting loss with the manifold smoothness loss, where the smoothness loss requires the classifier varies smoothly with respect to the local data manifold distribution. Then,the importance of each unlabeled data point can be automatically obtained through the corresponding output of the cost function. Instead of utilizing all training samples simultaneously to train the classifier, in each iteration, important samples for the current classifier can be automatically pseudo-labeled with their current predictions and then added into the training data in the following model training process. This provides the classifier with more reliable training data. With the sample importance evaluation and pseudo-labeling strategies, the class labels are propagated from labeled samples to unlabeled samples in a self-paced fashion. Finally, the alternative optimization strategy is utilized to obtain the explicit nonlinear multi-class classification function.

The main contributions of the paper are summarized as follows:

(1) The proposed SSPL method is able to guide the learning process through providing samples with importance values. The method pays more attention on the reliable patterns (with high importance) rather than the indistinctive ones (with less importance). Therefore, the classifier is robust to data with heavy noise and the outliers.
(2) Importance evaluation is key to the proposed SSPL. A new cost function for both labeled and unlabeled samples is defined as a mixture loss. It considers both the label predicting error and the smoothness with respect to the data manifold. The cost function can better describe the importance of the samples than any single loss, which efficiently extends self-paced learning to partially labeled training data.
(3) The importance of both labeled and unlabeled samples for classifier in the subsequent iteration can be determined adaptively by the proposed cost function, without need of manually designing.
(4) The proposed method is naturally inductive. The gained explicit nonlinear multi-class classification

function can be used to rapidly predict the labels of test samples.

The rest of the paper is organized as follows. Some related works are reviewed in Sect. 2. In Sect. 3, we introduce the structure regularized self-paced learning approach for semi-supervised classification. Experiments on benchmark real-world data sets are reported in Sect. 4. Finally, some concluding remarks are given in Sect. 5.

2 Background

2.1 Manifold regularization framework for semi-supervised classification

MR [15], a general framework for semi-supervised binary-classification problem, was proposed by Belkin et al. based on the regularization theory. In contrast with the traditional regularization theory which concentrates on the complexity of functions in a functional space, MR framework exploits the geometry of the probability distribution that generates data samples and incorporates it as an additional regularization term. In detail, the framework can be formulated as follows:

$$f^* = \arg\min_{f\in\mathcal{H}_K}\left\{\frac{1}{m}\sum_{i=1}^{m} L(z_i,f(\mathbf{x}_i)) + \gamma_K\|f\|_K^2 + \gamma_I\|f\|_I^2\right\}. \tag{1}$$

Here, f is the desired classification function, $\mathcal{H}_K$ is the reproducing kernel Hilbert space (RKHS), m is the number of labeled samples in the training set, $z_i \in \{-1,1\}(i = 1,\ldots,m)$ is the binary class label of the sample $\mathbf{x}_i$, $L(\cdot,\cdot)$ is certain loss function, $\|f\|_K^2$ is the complexity regularization term that measures the complexity of the classifier in $\mathcal{H}_K$, $\|f\|_I^2$ is the smoothness regularization term that measures the smoothness of the classifier with respect to the geometric distribution of data, γ_K and γ_I are two parameters.

Supposing that the data lie on a low-dimensional manifold embedded in high-dimensional space, the smoothness regularization term $\|f\|_I^2$ can be defined to measure the smoothness of the classifier with respect to the manifold geometry. To model the data manifold, usually a graph $\mathcal{G}$ on training data is utilized. Given a neighborhood size (integer k or positive real ε), there are usually two ways to construct the graph $\mathcal{G}$:

(1) *k-nearest neighborhood (k-NN) method*: If $\mathbf{x}_i$ is one of the k nearest data points to $\mathbf{x}_j$, set an edge between data points $\mathbf{x}_i$ and $\mathbf{x}_j$;

(2) *ε-neighborhood (ε-NN) method*: If $\|\mathbf{x}_i - \mathbf{x}_j\|_2 < \varepsilon$ where $\|\cdot\|_2$ is the 2-norm of vector, set an edge between data points $\mathbf{x}_i$ and $\mathbf{x}_j$.

The pair-wise weights of the edges in graph $\mathcal{G}$ can be defined as

$$w_{ij} = \begin{cases} 1 \text{ or } \exp\{-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/\sigma\} & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where σ is a parameter. Then, the regularization term $\|f\|_I^2$ can be defined as $\|f\|_I^2 = \frac{1}{n^2}\sum_{i,j=1}^{n}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 w_{ij}$, where n is number of training data points. Meanwhile, if the loss function is defined as: $L(z,f(\mathbf{x})) = (z - f(\mathbf{x}))^2$, then the Laplacian Regularized Least Square Classifier (LapRLSC) [15] can be obtained.

For different choices of the loss function $L(\cdot,\cdot)$ and the smoothness regularization term $\|f\|_I^2$, different MR algorithms can be derived. Though MR framework was proposed mainly for semi-supervised learning, it can actually develop algorithms including unsupervised, semi-supervised, and fully supervised learning. It can also unify many of the graph-based semi-supervised classification algorithms by ignoring the complexity regularization term, which leads to that the framework only has the error term and the smoothness regularization term.

2.2 Self-paced learning

Inspired by the learning principle of human/animal, Bengio et. al. proposed the concept of curriculum learning (CL) [30], that is, training a learning machine with a predefined curriculum that can gradually involve samples into training from easy to complex. However, the curriculum design in CL turns out difficult in real applications. Therefore, based on the learning philosophy of CL, Kumar et al. promoted CL as a new concise model, named Self-Paced Learning (SPL) [22]. Different from CL, SPL learns the training data from easy to complex adaptively determined by the feedback of the learner itself.

Denote $L(z_i,f(\mathbf{x}_i;\mathbf{w}))$ as the loss between the ground truth label z_i and the estimated label $f(\mathbf{x}_i;\mathbf{w})$, where $\mathbf{w}$ represents the model parameter of the classification function f. Then,SPL model can be expressed as [23, 31, 32]:

$$\min_{\mathbf{v}\in[0,1]^m,\mathbf{w}} \sum_{i=1}^{m}\{v_i L(z_i,f(\mathbf{x}_i;\mathbf{w})) + g(v_i,\lambda)\}. \tag{3}$$

Here, $\mathbf{v} = [v_1, v_2, \ldots, v_m]^{\mathrm{T}}$ denote the weight variables reflecting the importance of the training samples, $g(\cdot,\cdot)$ is called the self-paced regularizer (SP-regularizer) determining the learning scheme, and λ is the age parameter of SP-regularizer that controls the learning pace of the model.

SPL utilizes alternative optimization strategy to jointly learn the model parameter $\mathbf{w}$ and the latent weight vector $\mathbf{v}$. By sequentially optimizing the model with gradually increasing age parameter, more and more samples can be automatically included into training from easy to complex in a pure self-paced way.

Jiang et al. [31, 32] have presented a formal definition for the self-paced regularizer $g(v,\lambda)$. That is, $g(v,\lambda)$ should satisfy: (1) $g(v,\lambda)$ is convex with respect to $v \in [0,1]$; (2) the optimal weight $v^*(\lambda,\ell) = \arg\min_{v\in[0,1]}(v\ell + g(v,\lambda))$ is monotonically decreasing with respect to the loss $\ell = L(z,f(\mathbf{x};\mathbf{w}))$, and it holds that $\lim_{\ell\to 0} v^*(\lambda,\ell) = 1$, $\lim_{\ell\to\infty} v^*(\lambda,\ell) = 0$; (3) the optimal weight $v^*(\lambda,\ell)$ is monotonically increasing with respect to λ, and it holds that $\lim_{\lambda\to\infty} v^*(\lambda,\ell) \le 1, \lim_{\lambda\to 0} v^*(\lambda,\ell) = 0$.

In this definition, condition (2) indicates that the model is inclined to select easy samples (with smaller loss) rather than complex samples (with larger loss); condition (3) states that when the model "age" λ gets larger, it tends to incorporate more, probably complex, samples to train a "mature" model; the convexity in condition (1) further ensures the soundness of the regularizer for optimization. Under this definition, different SP-regularizers can be constructed, such as the hard weighting regularizer, linear soft weighting regularizer, logarithmic weighting regularizer, and the mixture weighting regularizer [22, 31, 32].

Alternative optimization strategy (AOS) [22, 33] is generally utilized to solve problem (3). AOS is an iterative method for optimization, which divides the variables into a set of disjoint blocks and optimizes each block of variables while keeping other blocks fixed in each iteration. For problem (3), when $\mathbf{v}$ is fixed, we can utilize the existing off-the-shelf supervised learning methods to obtain the optimal $\mathbf{w}$. When $\mathbf{w}$ is fixed, taking the hard SP-regularizer $g^H(v_i,\lambda) = -\lambda v_i$ for example, the global optimum $\mathbf{v}^* = [v_1^*, v_2^*, \ldots, v_m^*]^{\mathrm{T}}$ can be easily obtained by [22]:

$$v_i^* = \begin{cases} 1, & \text{if } L(z_i,f(\mathbf{x}_i;\mathbf{w})) < \lambda \\ 0, & \text{if } L(z_i,f(\mathbf{x}_i;\mathbf{w})) \ge \lambda \end{cases} \tag{4}$$

There exists an intuitive explanation behind this alternative optimization strategy. On the one hand, when updating $\mathbf{v}$ with a fixed $\mathbf{w}$, the sample whose loss $L(z_i,f(\mathbf{x}_i;\mathbf{w}))$ is smaller than the parameter λ is taken as an "easy" sample ($v_i^* = 1$), and will be incorporated into the training process for the next iteration, or otherwise unincorporated ($v_i^* = 0$). On the other hand, when updating $\mathbf{w}$ with a fixed $\mathbf{v}$, the classifier is trained only on the selected "easy" samples. The parameter λ controls the pace at which the model learns new samples, and physically λ corresponds to the "age" of the learner. At the beginning of the training, the age parameter λ is set to be small, and in this way only the "easy" samples with small losses can be taken into account; then, with the growing of λ, more samples with larger losses will be incorporated to train a more "mature" model [24].

Based on the SPL learning regime, multiple variations have been proposed [24–26, 31, 32, 34]. For example, [34] proposed a unified framework named self-paced curriculum learning, that can make use of both prior knowledge before training and dynamical information extracted during training, and stated that this regime is analogous to an "instructor-student-collaborative" leaning mode. In [24], Jiang et al. proposed an approach called self-paced learning with diversity, which not only prefers easy samples but also diverse samples in training. In [26], Lin et. al. combined active learning with SPL and introduced a novel cost-effective framework for face identification, which builds classifiers by progressively annotating and selecting unlabeled samples in an active self-paced way.

3 The structure regularized self-paced learning method

In this section, we introduce the details of the proposed SSPL method for semi-supervised classification. We first present the mathematical formulation of the proposed model and then introduce the alternative optimization algorithm for solving this model.

3.1 Structure regularized self-paced learning model

For semi-supervised classification problem, we denote the partially labeled training data set as $\mathcal{D} = \{(\mathbf{x}_1,\mathbf{z}_1),\ldots,(\mathbf{x}_m,\mathbf{z}_m),\ \mathbf{x}_{m+1},\ldots,\mathbf{x}_n\}$. Here, m is the number of labeled samples, n is the number of total training samples, $\mathbf{x}_i$ is a D-dimensional feature representation for the ith sample. For data sets with C classes, we denote $\mathbf{z}_i = [z_i^1,\ldots,z_i^C]^{\mathrm{T}} \in \mathbb{R}^C$ as the class label for $\mathbf{x}_i$, where z_i^j corresponds to the label of $\mathbf{x}_i$ to the jth class. That is, if $\mathbf{x}_i$ belongs to the kth ($k = 1,2,\ldots,C$) class, then $z_i^k = 1$ and $z_i^j = 0 (j = 1,\ldots,C, j \ne k)$. All data points and the corresponding label vectors are in the form of column vectors and denoted by bold lowercases. Matrices are denoted by capital bold letters.

We first give an overview of the proposed method, which iterates among classifier updating, sample importance calculating and pseudo-labeling in a self-paced fashion.

(1) *Classifier updating* We consider three terms for classifier training, that is, the label fitting loss of the labeled and the pseudo-labeled samples, the functional complexity, and the smoothness with respect

to the data manifold. In the beginning, only the labeled samples are utilized for training; then, with the self-paced learning regime, more unlabeled samples are pseudo-labeled and incorporated into training.

(2) *Sample importance calculating* In each iteration, once the classifier has been updated, the mixture losses of unlabeled samples can be computed, taking account of both the pseudo-label fitting loss and the smoothness of the classifier with respect to the locally linear reconstruction error. Then, the importance values of the unlabeled samples can be obtained.

(3) *Pseudo-labeling* After the calculation of the importance values of the unlabeled samples, we can assign or re-assign the pseudo-labels to samples with high importance values. As iteration goes on, the ground truth labels and pseudo-labels can be propagated smoothly from labeled samples to unlabeled samples in the self-paced manner.

The general formulation of the proposed structure regularized self-paced label propagation method is presented as follows.

$$\begin{aligned}\min_{f_s\in\mathcal{H}_K,\mathbf{v},\{\mathbf{z}_i\}_{i=m+1}^n}\Bigg\{&\frac{1}{n}\sum_{i=1}^n v_i L(\mathbf{z}_i,\mathbf{f}(\mathbf{x}_i))+\gamma_K\|\mathbf{f}\|_K^2\\&+\frac{\gamma_I}{n}\sum_{i=1}^n v_i L_I(\mathbf{x}_i)+\frac{1}{n}\sum_{i=1}^n g(v_i,\lambda)\Bigg\}\\&\text{s.t.}\quad v_i\in\Psi_i^\lambda\quad(i=1,\ldots,n)\end{aligned}\tag{5}$$

Here, $\mathbf{f}=[f_1,f_2,\ldots,f_C]^{\mathrm{T}}$ is the desired multi-class classification function, $\mathbf{v}=[v_1,v_2,\ldots,v_n]^{\mathrm{T}}$ is the weight variables reflecting the importance of the training samples, $\mathbf{z}_i$ is the ground truth label for the labeled sample $\mathbf{x}_i(i=1,2,\ldots,m)$ or pseudo-label for the unlabeled sample $\mathbf{x}_i(i=m+1,m+2,\ldots,n)$, $L(\cdot,\cdot)$ is the loss function, $\mathcal{H}_K$ is certain RHKS, γ_K and γ_I are regularization parameters, $\|\cdot\|_K$ is the norm of the function in RKHS $\mathcal{H}_K$, $L_I(\mathbf{x}_i)$ is the smoothness loss of the function $\mathbf{f}$ at the data manifold around $\mathbf{x}_i$, $g(\cdot,\cdot)$ is the self-paced regularizer and λ is the corresponding age parameter, $\cap_{i=1}^n\{v_i\in\Psi_i^\lambda\}$ is the predetermined curriculum constraint of the model at the pace age λ [26, 34].

Roughly speaking, in the optimization problem (5), the first term of the objective function measures the weighted average loss between the predicted label $\mathbf{f}(\mathbf{x}_i)$ and the ground truth label or the pseudo-label $\mathbf{z}_i$; the second term of the objective function is the complexity regularization term that controls the complexity of the classifier in the ambient space; the third term of the objective function is the smoothness regularization term that controls the smoothness of the classifier with respect to the data manifold structure; the last term of the objective function is the self-paced regularizer that controls the self-paced learning scheme of the classifier; the optimization constraint imposes a prior knowledge about the sample importance for the self-paced learning scheme.

For the first term of the objective function of problem (5), we simply define the fitting loss function as:

$$L(\mathbf{z}_i,\mathbf{f}(\mathbf{x}_i))=\|\mathbf{z}_i-\mathbf{f}(\mathbf{x}_i)\|_2^2,\tag{6}$$

where $\|\cdot\|_2$ is the 2-norm of vector.

For the second term of the objective function (5), we define $\|\mathbf{f}\|_K^2$ as the square of the norm of $\mathbf{f}$ in the ambient space. It is known that any positive semi-definite kernel $k(\cdot,\cdot)$ gives rise to an RKHS $\mathcal{H}_K$, which can be constructed by considering the space of finite linear combinations of kernels $\sum_i\eta_i k(\mathbf{x}_i,\cdot)$ with the inner product being $\langle k(\mathbf{x}_i,\cdot),k(\mathbf{x}_j,\cdot)\rangle_{\mathcal{H}_K}=k(\mathbf{x}_i,\mathbf{x}_j)$ [15, 35]. Then, for the function $f_s\in\mathcal{H}_K$, we can define $\|f_s\|_K^2$ as the square of the norm of f_s in $\mathcal{H}_K$. For the vector function $\mathbf{f}=[f_1,f_2,\ldots,f_C]^{\mathrm{T}}$, $\|\mathbf{f}\|_K^2$ can be defined as the summation of the norms of all component functions:

$$\|\mathbf{f}\|_K^2=\sum_{s=1}^C\|f_s\|_K^2.\tag{7}$$

For the third term of the objective function of problem (5), we make use of locally linear reconstruction method to measure the smoothness of the classifier with respect to the data manifold distribution. Supposing that each data point and its neighbors lie on or close to a locally linear patch of the manifold, we can characterize the local geometry of these patches by linear coefficients that reconstruct each data point from its neighbors. Then the overlapped locally linear patches can well discover the global nonlinear manifold structure [36]. For each training data point $\mathbf{x}_i(i=1,2,\ldots,n)$, the coefficients of $\mathbf{x}_j$ for the locally linear reconstruction of $\mathbf{x}_i$ can be obtained by solving the following problem:

$$\begin{aligned}&\min_{M_{ip}}\quad\|\mathbf{x}_i-\sum_{p=1}^n M_{ip}\mathbf{x}_p\|_2^2\\&\text{s.t.}\ \sum_{p=1}^n M_{ip}=1,\quad\text{and}\quad M_{ip}=0\quad(\text{if }\mathbf{x}_p\notin\mathcal{N}_i),\end{aligned}\tag{8}$$

where $\mathcal{N}_i=\{\mathbf{x}_{i_1},\mathbf{x}_{i_2},\ldots,\mathbf{x}_{i_k}\}$ is the k-nearest neighborhood set or ε-neighborhood set of $\mathbf{x}_i$. The optimal weights of problem (8) can be computed in closed form, as stated in Proposition 1. Supposing that the classification function $\mathbf{f}$ is approximately locally linear, we expect that the label of $\mathbf{x}_i$ can also be approximately locally linear reconstructed by the labels of the nearby neighbors, utilizing the same reconstruction coefficients. Therefore, we define

$L_I(\mathbf{x}_i)$, the smoothness loss of the classification function $\mathbf{f}$ at the data manifold around $\mathbf{x}_i$, as the following reconstruction error:

$$L_I(\mathbf{x}_i) = \frac{1}{n}\|\mathbf{f}(\mathbf{x}_i) - \sum_{p=1}^{n} M_{ip}\mathbf{f}(\mathbf{x}_p)\|_2^2. \tag{9}$$

Proposition 1 *The optimal weights of problem (8) are:*

$$\mathbf{w}_i = \mathbf{C}^{-1}\mathbf{1}/\alpha. \tag{10}$$

Here, $\mathbf{w}_i = [M_{i,i_1}, M_{i,i_2}, \ldots, M_{i,i_k}]^{\mathrm{T}} \in \mathbb{R}^k$ *represent the coefficients of* $\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_k}$ *for the reconstruction of* $\mathbf{x}_i$, $\mathbf{C} = (c_{qj})_{k\times k}$ *is the Gram matrix with* $c_{qj} = \boldsymbol{\eta}_q^{\mathrm{T}}\boldsymbol{\eta}_j$ *and* $\boldsymbol{\eta}_q = \mathbf{x}_i - \mathbf{x}_{i_q}$, $\mathbf{1} = [1, 1, \ldots, 1]^{\mathrm{T}}$ *is a k-dimensional vector,* α *is the sum of all elements in* $\mathbf{C}^{-1}$.

Proof The objective function of problem (8) can be transformed as:

$$\|\mathbf{x}_i - \sum_{p=1}^{n} M_{ip}\mathbf{x}_p\|_2^2 = \|\sum_{p=1}^{n} M_{ip}(\mathbf{x}_i - \mathbf{x}_p)\|_2^2 = \|\sum_{q=1}^{k} M_{i,i_q}\boldsymbol{\eta}_q\|_2^2. \tag{11}$$

Therefore, we can define the Lagrange function as

$$\mathcal{L} = \|\sum_{q=1}^{k} M_{i,i_q}\boldsymbol{\eta}_q\|_2^2 - \tilde{\alpha}\left(\sum_{q=1}^{k} M_{i,i_q} - 1\right) \tag{12}$$

where $\tilde{\alpha}$ is the Lagrange multiplier. Let $\frac{\partial \mathcal{L}}{\partial M_{i,i_q}} = 2\sum_{j=1}^{k} M_{i,i_j}\boldsymbol{\eta}_q^{\mathrm{T}}\boldsymbol{\eta}_j - \tilde{\alpha} = 0$, we have $\sum_{j=1}^{k} M_{i,i_j}\boldsymbol{\eta}_q^{\mathrm{T}}\boldsymbol{\eta}_j = \frac{\tilde{\alpha}}{2}$, $(q = 1, 2, \ldots, k)$. Therefore, $C\mathbf{w}_i = \frac{\tilde{\alpha}}{2}\mathbf{1}$, $\mathbf{w}_i = \frac{\tilde{\alpha}}{2}C^{-1}\mathbf{1}$. Besides, from $\mathbf{1}^{\mathrm{T}}\mathbf{w}_i = 1$ we can know that $\frac{\tilde{\alpha}}{2} = \frac{1}{\alpha}$. Then, the conclusion of this proposition can be got. □

Remark 1 In Eq. (10), if the matrix $\mathbf{C}$ is nearly singular, we can add a small multiple of the identity matrix to $\mathbf{C}$.

The last term of the objective function of problem (5) is the self-paced regularizer. Similar to the scheme that human learns knowledge, self-paced regularizer determines a scheme for the model to learn new samples. According to the cost value of each sample, one can define different kinds of self-paced regularizer to assign different kinds of sample importance calculating regime, such as hard weighting regime that provides the importance value $v \in \{0, 1\}$ and the soft weighting regime that provides $v \in [0, 1]$. In this paper, we utilize the following linear soft weighting regularizer since it is easy to implement and is robust to complex data sets:

$$g(v_i, \lambda) = \frac{\lambda}{2}(v_i^2 - 2v_i). \tag{13}$$

For the optimization constraint $v_i \in \Psi_i^{\lambda}(i = 1, \ldots, n)$, $\bigcap_{i=1}^{n}\{v_i \in \Psi_i^{\lambda}\}$ is the predetermined curriculum that weakly guides the learning from easy to complex samples.

The curriculum can be seen as a training procedure that is associated with a set of weights on training samples, or more generally, on a reweighting of the training data distribution. Specifically, here we set Ψ_i^{λ}, the curriculum constraint for sample $\mathbf{x}_i$, as following:

(1) *The curriculum for labeled samples* For each labeled sample $\mathbf{x}_i(i = 1, \ldots, m)$, we set $\Psi_i^{\lambda} = \{1\}$. That is, the importance values of the labeled samples are fixed as $v_i = 1$ during the training process. In this way, the discriminative information hidden in the labeled training data can be fully investigated.

(2) *The curriculum for unlabeled samples* For each unlabeled sample $\mathbf{x}_i(i = m + 1, \ldots, n)$, we set $\Psi_i^{\lambda} = [0, 1]$. The importance value of $\mathbf{x}_i$ is learned in the self-paced learning procedure, depending on the value of cost of the sample by the current classifier.

This definition of curriculum can be considered as an "instructor-student collaborative" learning mode, as opposed to "student driven" learning mode in SPL and "instructor driven" learning mode in previous curriculum learning works [26, 34]. With this curriculum, instructors provide prior knowledge on a weak learning sequence of samples, while leaving the learner some freedom to adjust to the actual curriculum according to the learning pace.

In summary, based on the above discussions, the proposed SSPL model can be formulated as:

$$\begin{aligned}\min_{f_s\in\mathcal{H}_K,\mathbf{v},\{\mathbf{z}_i\}_{i=m+1}^{n}} &\left\{\frac{1}{n}\sum_{i=1}^{n} v_i\|\mathbf{z}_i - \mathbf{f}(\mathbf{x}_i)\|_2^2 + \gamma_K\sum_{s=1}^{C}\|f_s\|_K^2\right.\\ &\left.+\frac{\gamma_I}{n^2}\sum_{i=1}^{n} v_i\|\mathbf{f}(\mathbf{x}_i) - \sum_{p=1}^{n} M_{ip}\mathbf{f}(\mathbf{x}_p)\|_2^2 + \frac{\lambda}{2n}\sum_{i=1}^{n}(v_i^2 - 2v_i)\right\}\\ \text{s.t.}\quad & v_i \in \Psi_i^{\lambda}\quad (i = 1, \ldots, n)\end{aligned} \tag{14}$$

3.2 Alternative optimization strategy for the SSPL model

We can make use of alternative optimization strategy [22, 33] to solve the proposed SSPL problem (14). AOS is an iterative method for solving optimization problem. It divides the variables into a set of disjoint blocks and optimizes each block alternatively in each iteration. In the case of problem (14), the variables are divided into three blocks: the classifier parameters, importance values $\mathbf{v}$ of samples, and the pseudo-labels $\mathbf{z}_i(i = m + 1, \ldots, n)$. In detail, the AOS process for the proposed SSPL problem (14) is presented as follows:

(1) *Initialization* In this step we set the initial values of parameters of the algorithm. For importance values of samples, we set $v_i = 1$ for labeled samples $\mathbf{x}_i (i = 1, \ldots, m)$, and $v_i = 0$ for unlabeled samples $\mathbf{x}_i (i = m+1, \ldots, n)$. The age parameter λ is initialized with a small value to allow only labeled samples into training for the first iteration. The regularization parameters γ_k and γ_I are fixed as specific values during the training process.

(2) *Classifier updating* This step is to optimize the parameters of the classifier with fixed importance weights $\mathbf{v}$ and pseudo-labels of important samples. In this case, the SSPL problem (14) can be reformulated as follows:

$$\min_{f_s \in \mathcal{H}_K} \left\{ \frac{1}{n} \sum_{i=1}^{n} v_i \|\mathbf{z}_i - \mathbf{f}(\mathbf{x}_i)\|_2^2 + \gamma_K \sum_{s=1}^{C} \|f_s\|_K^2 + \frac{\gamma_I}{n^2} \sum_{i=1}^{n} v_i \|\mathbf{f}(\mathbf{x}_i) - \sum_{p=1}^{n} M_{ip} \mathbf{f}(\mathbf{x}_p)\|_2^2 \right\}. \tag{15}$$

For this problem, we have the following representer theorem, showing that the minimizer has an expansion in terms of both labeled and unlabeled samples.

Theorem 1 (Representer Theorem) *The minimizer of problem* (15) *admits an expansion*

$$\mathbf{f}(\mathbf{x}) = \sum\nolimits_{i=1}^{n} \mathbf{b}_i k(\mathbf{x}_i, \mathbf{x}), \tag{16}$$

where $\mathbf{b}_i = [b_{1i}, \ldots, b_{Ci}]^{\mathrm{T}} \in \mathbb{R}^C$.

Proof The theorem can be proved similarly as Theorem 2 of [15], which states that the LapRLSC problem:

$$\arg\min_{f \in \mathcal{H}_K} \left\{ \frac{1}{m} \sum_{i=1}^{m} L(z_i, f(\mathbf{x}_i)) + \gamma_K \|f\|_K^2 + \frac{\gamma_I}{n^2} \sum_{i,j=1}^{n} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 w_{ij} \right\} \tag{17}$$

admits an expansion $f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) \in \mathbb{R}$ and $\alpha_i \in \mathbb{R}$, where $f(\cdot)$ is the classification function for binary-classification problem.

In the following, we will give the proof of Theorem 1. For the RKHS $\mathcal{H}_K$ corresponding to the kernel $k(\cdot, \cdot)$, any function f_s in $\mathcal{H}_K$ can be uniquely decomposed into a component $(f_s)_{\|}$ in the linear subspace spanned by the kernel functions $\{k(\mathbf{x}_i, \cdot)\}_{i=1}^{n}$ and a component $(f_s)_{\perp}$ orthogonal to it. Thus,

$$f_s = (f_s)_{\|} + (f_s)_{\perp} = \sum_{i=1}^{n} b_{si} k(\mathbf{x}_i, \cdot) + (f_s)_{\perp}. \tag{18}$$

Then, we have $\mathbf{f} = [f_1, \ldots, f_C]^{\mathrm{T}} = \sum_{i=1}^{n} \mathbf{b}_i k(\mathbf{x}_i, \cdot) + \mathbf{f}_{\perp}$, where $\mathbf{b}_i = [b_{1i}, \ldots, b_{Ci}]^{\mathrm{T}} \in \mathbb{R}^C$ and $\mathbf{f}_{\perp} = [(f_1)_{\perp}, \ldots, (f_C)_{\perp}]^{\mathrm{T}} \in \mathbb{R}^C$.

For any training sample $x_j (j = 1, 2, \ldots, n)$, we have

$$\begin{aligned} \mathbf{f}(\mathbf{x}_j) &= [\langle f_1, k(\mathbf{x}_j, \cdot)\rangle, \ldots, \langle f_C, k(\mathbf{x}_j, \cdot)\rangle]^{\mathrm{T}} \\ &= \left[\left\langle \sum_{i=1}^{n} b_{1i} k(\mathbf{x}_i, \cdot), k(\mathbf{x}_j, \cdot) \right\rangle + \langle (f_1)_{\perp}, k(\mathbf{x}_j, \cdot)\rangle, \ldots, \left\langle \sum_{i=1}^{n} b_{Ci} k(\mathbf{x}_i, \cdot), k(\mathbf{x}_j, \cdot) \right\rangle + \langle (f_C)_{\perp}, k(\mathbf{x}_j, \cdot)\rangle \right]^{\mathrm{T}}. \end{aligned} \tag{19}$$

Since $\langle (f_s)_{\perp}, k(\mathbf{x}_j, \cdot)\rangle = 0$ and $\langle k(\mathbf{x}_i, \cdot), k(\mathbf{x}_j, \cdot)\rangle = k(\mathbf{x}_i, \mathbf{x}_j)$, we can get that

$$\mathbf{f}(\mathbf{x}_j) = \sum_{i=1}^{n} \mathbf{b}_i k(\mathbf{x}_i, \mathbf{x}_j). \tag{20}$$

This means that $\mathbf{f}(\mathbf{x}_j)$ is independent of the orthogonal component $\mathbf{f}_{\perp}$. Therefore, the first and third terms of the optimization function in (15) are independent of the orthogonal component $\mathbf{f}_{\perp}$. In other words, the value of $\mathbf{f}_{\perp}$ will not affect the values of the first and third terms.

In fact, the orthogonal component $\mathbf{f}_{\perp}$ only increases the complexity regularization term $\sum_{s=1}^{C} \|f_s\|_K^2$, since

$$\|f_s\|_K^2 = \|\sum_{i=1}^{n} b_{si} k(\mathbf{x}_i, \cdot)\|_K^2 + \|(f_s)_{\perp}\|_K^2 \geq \|\sum_{i=1}^{n} b_{si} k(\mathbf{x}_i, \cdot)\|_K^2. \tag{21}$$

Thus, the minimizer of the problem (15) must have a zero orthogonal component $\mathbf{f}_{\perp} = 0$ and we can see that the solution of problem (15) admits a representation $\mathbf{f}(\cdot) = \sum_{i=1}^{n} \mathbf{b}_i k(\mathbf{x}_i, \cdot)$.

Substituting the expansion (16) into (15), we can get the following matrix formulations:

$$\begin{aligned} &\sum_{i=1}^{n} v_i \|\mathbf{z}_i - \mathbf{f}(\mathbf{x}_i)\|_2^2 = \mathrm{tr}\Big((\mathbf{Z} - \mathbf{BK})\mathbf{V}(\mathbf{Z} - \mathbf{BK})^{\mathrm{T}}\Big) \\ &\|f_s\|_K^2 = \boldsymbol{\beta}_s K \boldsymbol{\beta}_s^T \qquad \sum_{s=1}^{C} \|f_s\|_K^2 = \mathrm{tr}(\mathbf{BKB}^{\mathrm{T}}) \\ &\sum_{i=1}^{n} v_i \|\mathbf{f}(\mathbf{x}_i) - \sum_{p=1}^{n} M_{ip} \mathbf{f}(\mathbf{x}_p)\|_2^2 = \mathrm{tr}\big(\mathbf{BK}(\mathbf{I} - \mathbf{M})^T \mathbf{V}(\mathbf{I} - \mathbf{M})\mathbf{KB}^{\mathrm{T}}\big) \end{aligned} \tag{22}$$

Here, $\mathrm{tr}(\cdot)$ is the trace operator of a matrix, $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n] \in \mathbb{R}^{C \times n}$ represents the label matrix, $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{R}^{C \times n}$ is the coefficient matrix, $\mathbf{K} = (k(\mathbf{x}_i, \mathbf{x}_j)) \in \mathbb{R}^{n \times n}$ is the kernel matrix, $\mathbf{V} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with the ith $(i = 1, 2, \ldots, n)$ diagonal element being v_i, $\boldsymbol{\beta}_s$ is the sth row of matrix $\mathbf{B}$, $\mathbf{I}$ is the identity matrix of size n, $\mathbf{M} = (M_{ij}) \in \mathbb{R}^{n \times n}$ is the locally linear reconstruction coefficient matrix.

Therefore, the problem (15) can be reformulated as follows.

$$\min_{\mathbf{B}}\Big\{\frac{1}{n}\mathrm{tr}\Big((\mathbf{Z}-\mathbf{BK})\mathbf{V}(\mathbf{Z}-\mathbf{BK})^{\mathrm{T}}\Big)+\gamma_K\mathrm{tr}(\mathbf{BKB}^{\mathrm{T}}) +\frac{\gamma_I}{n^2}\mathrm{tr}\Big(\mathbf{BK}(\mathbf{I}-\mathbf{M})^{\mathrm{T}}\mathbf{V}(\mathbf{I}-\mathbf{M})\mathbf{KB}^{\mathrm{T}}\Big)\Big\}. \tag{23}$$

Forcing the derivative of the objective function with respect to $\mathbf{B}$ being 0, we can get the optimal solution:

$$\mathbf{B}^*=\mathbf{ZV}\Big(\mathbf{KV}+\gamma_K n\mathbf{I}+\frac{\gamma_I}{n}\mathbf{K}(\mathbf{I}-\mathbf{A})^T\mathbf{V}(\mathbf{I}-\mathbf{A})\Big)^{-1}. \tag{24}$$

Then, the optimal classification function of problem (15) is obtained as

$$\mathbf{f}^*(\mathbf{x})=[f_1^*(\mathbf{x}),f_2^*(\mathbf{x}),\ldots,f_C^*(\mathbf{x})]^{\mathrm{T}}=\sum_{i=1}^{n}\mathbf{b}_i^*k(\mathbf{x}_i,\mathbf{x}), \tag{25}$$

where $\mathbf{b}_i^*$ is the ith column of $\mathbf{B}^*$. Correspondingly, the classifier is updated as

$$\mathrm{identity}(\mathbf{x})=s^*=\arg\max_{s=1,2,\ldots,C}\{f_s^*(\mathbf{x})\}. \tag{26}$$

(3) *Sample importance calculating* The purpose of this step is to assign importance vales v_i to the training samples, and further pick the important unlabeled samples (with nonzero values $\mathbf{v}_i$) for the training process at the next iteration. In this case, with fixed classifier $\mathbf{f}$ and pseudo-labels of important samples, the SSPL problem (14) turns out to be the following optimization problem:

$$\begin{aligned}\min_{\mathbf{v}}\Big\{&\sum_{i=1}^{n}v_i\|\mathbf{z}_i-\mathbf{f}(\mathbf{x}_i)\|_2^2+\frac{\gamma_I}{n}\sum_{i=1}^{n}v_i\|\mathbf{f}(\mathbf{x}_i)\\&-\sum_{p=1}^{n}M_{ip}\mathbf{f}(\mathbf{x}_p)\|_2^2+\frac{\lambda}{2}\sum_{i=1}^{n}(v_i^2-2v_i)\Big\}\\ \text{s.t.}\quad &v_i\in\Psi_i^{\lambda}\quad(i=1,\ldots,n)\end{aligned} \tag{27}$$

This optimization problem is separable with respect to v_i. For labeled samples $\mathbf{x}_i(i=1,\ldots,m)$, the optimal solution is $v_i^*=1$. For unlabeled samples $\mathbf{x}_i(i=m+1,\ldots,n)$, the optimal solution can be gained by solving the following problem:

$$\begin{aligned}\min_{v_i\in[0,1]}\mathcal{E}(\mathbf{v})=\Big\{&v_i\|\mathbf{z}_i-\mathbf{f}(\mathbf{x}_i)\|_2^2+\frac{\gamma_I}{n}v_i\|\mathbf{f}(\mathbf{x}_i)\\&-\sum_{p=1}^{n}M_{ip}\mathbf{f}(\mathbf{x}_p)\|_2^2+\frac{\lambda}{2}(v_i^2-2v_i)\Big\}.\end{aligned} \tag{28}$$

The objective function $\mathcal{E}(\mathbf{v})$ is convex with respect to variable $\mathbf{v}$, and the global minimum can be obtained by forcing the derivative for v_i to be 0:

$$\begin{aligned}\frac{\partial\mathcal{E}(\mathbf{v})}{\partial v_i}=&\lambda v_i+\|\mathbf{z}_i-\mathbf{f}(\mathbf{x}_i)\|_2^2+\frac{\gamma_I}{n}\|\mathbf{f}(\mathbf{x}_i)\\&-\sum_{p=1}^{n}M_{ip}\mathbf{f}(\mathbf{x}_p)\|_2^2-\lambda=0.\end{aligned} \tag{29}$$

Considering $v_i\in[0,1]$, the close-formed optimal solution for problem (27) can be obtained:

$$v_i^*=\begin{cases}1, & i=1,\ldots,m\\ \begin{cases}1-C(\mathbf{x}_i)/\lambda & C(\mathbf{x}_i)<\lambda\\ 0 & C(\mathbf{x}_i)\geq\lambda\end{cases}, & i=m+1,\ldots,n\end{cases} \tag{30}$$

where the new cost function consists of a mixture of losses as

$$C(\mathbf{x}_i)=\|\mathbf{z}_i-\mathbf{f}(\mathbf{x}_i)\|_2^2+\frac{\gamma_I}{n}\|\mathbf{f}(\mathbf{x}_i)-\sum_{p=1}^{n}M_{ip}\mathbf{f}(\mathbf{x}_p)\|_2^2. \tag{31}$$

The new cost function defined in Eq. (31) combines the label fitting error $L(\mathbf{z}_i,\mathbf{f}(\mathbf{x}_i))=\|\mathbf{z}_i-\mathbf{f}(\mathbf{x}_i)\|_2^2$ with the smoothness term $L_I(\mathbf{x}_i)=\frac{1}{n}\|\mathbf{f}(\mathbf{x}_i)-\sum_{p=1}^{n}M_{ip}\mathbf{f}(\mathbf{x}_p)\|_2^2$ of the classier with respect to the local data manifold distribution. If the regularization parameter γ_I is small, the label fitting loss that measures the difference between pseudo-label and predicted label will dominate the mixture loss; if the parameter γ_I is large, the smoothness loss that measures the locally linear reconstruction ability of $\mathbf{f}$ will dominate the mixture loss.

The mixture loss (31) measures the learning easiness of the unlabeled sample for the current classifier. The unlabeled sample with both low label fitting loss (guaranteeing the consistency of label predicting) and low smoothness loss (guaranteeing the smoothness of label propagating) can be seen as easy/important samples. In other words, this step examines the easiness of each sample based on what it has already learned, and adaptively determines their importance values to be used in the subsequent iterations. Besides, it can be seen from Eq. (30) that only the labeled samples and important unlabeled samples can be incorporated into training at the next iteration.

(4) *Pseudo-labeling of unlabeled samples* With fixed classifier $\mathbf{f}$ and importance value $\mathbf{v}$, the SSPL problem (14) is equivalent with the following optimization problem: $\min_{\mathbf{z}_i\in[0,1]^C}\sum_{i=m+1}^{n}v_i\|\mathbf{z}_i-\mathbf{f}(\mathbf{x}_i)\|_2^2$. For unlabeled samples $\mathbf{x}_i$ with nonzero importance value v_i, its pseudo-label can be deduced by the sub-problem: $\min_{\mathbf{z}_i\in[0,1]^C}\|\mathbf{z}_i-\mathbf{f}(\mathbf{x}_i)\|_2^2$. Therefore, we can assign the pseudo-label of $\mathbf{x}_i$ as:

$$z_i^s = \begin{cases} 0 & \text{if} \quad f_i^s \le 0 \\ f_i^s & \text{if} \quad 0 \le f_i^s \le 1 \\ 1 & \text{if} \quad f_i^s \ge 1 \end{cases}, \qquad (s = 1, 2, \ldots, C) \tag{32}$$

where $\mathbf{z}_i = [z_i^1, z_i^2, \ldots, z_i^C]^{\mathrm{T}}$ is the pseudo-label of $\mathbf{x}_i$, and $\mathbf{f}(\mathbf{x}_i) = [f_i^1, f_i^2, \ldots, f_i^C]^{\mathrm{T}}$ is the predicted label of $\mathbf{x}_i$ by the current classifier.

Once pseudo-labels of important unlabeled samples have been assigned, the self-paced age parameter λ is enlarged to allow more samples with larger mixture loss values into training in the next iteration, and then we repeat the above optimization process with respect to each variable. In specific, to update λ, we can specify the number of unlabeled samples to be included in each iteration, and then calculate λ according to Eq. (30). For example, if p unlabeled samples are needed to be selected for the current iteration; we first sort the unlabeled samples in ascending order of their mixture loss values, and then set λ as the loss value of $(p+1)$th sample.

Algorithm 1 AOS algorithm for SSPL

Input: The training data set $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{z}_1), \cdots, (\mathbf{x}_m, \mathbf{z}_m), \mathbf{x}_{m+1}, \cdots, \mathbf{x}_n\}$, two regularization parameters γ_K and γ_I, the kernel function $k(\cdot, \cdot)$;

Output: The classifier $\mathbf{f}$.

1: Initialize the importance weights $\{v_1, \cdots, v_n\}$;
2: **while** not converge & not all unlabeled samples are pesudo-labeled **do**
3: Update the classifier via (24) and (25);
4: Update the sample importance weights $\{v_1, \cdots, v_n\}$ by (30);
5: Compute pseudo-labels of important unlabeled samples by (32);
6: Renew the age parameter λ.
7: **end while**
8: Update the classifier via (24) and (25);
9: return the classification function $\mathbf{f}$.

The entire AOS algorithm for the proposed SSPL method is summarized in Algorithm 1. Such an algorithm converges since the objective function is monotonically decreasing and is bounded from below. In detail, it can be seen from Algorithm 1 that the method alternatively updates three batches of variables: the classifier parameters $\mathbf{B}$, the importance weights $\mathbf{v}$, and the pesudo-labels $\mathbf{z}_i$. These updates are deduced by a global optimum obtained from a sub-problem of the original model, then the objective can be guaranteed to decrease.

In Algorithm 1, initially the age parameter λ is set to be a small value, then λ is enlarged in each iteration. Besides, from Eq. (30) we can see that, in each iteration the model tends to emphasize and select the labeled samples and easy/important unlabeled samples that have smaller mixture losses than the current age parameter λ. Therefore, at the beginning of the training process, since the age parameter λ of self-paced regularizer is small, only labeled samples and the high-confidence/easy unlabeled samples are emphasized and selected for training; then, by sequentially optimizing the model with gradually increasing age parameter λ, more and more samples, that are probably more complex, can be utilized for training in a pace adaptively controlled by what it has already learned. With this self-controlled sample selection regime, SSPL smoothly guides the learning to emphasize the patterns of the reliable discriminative samples rather than those confusing ones. In this way, SSPL trains a more and more "mature" model, and thus can obtain both effective and robust learning performance.

(5) *Complexity analysis* The time complexity is a crucial issue in applications. The nearest neighbor graph construction process, including the locally linear coding, needs $O(kn^2)$ computational time. Subsequently, the classifier updating step (24) computes the inversion of a matrix, which consumes $O(n^3)$ time. One requires $O(n^2)$ computational time to evaluate the predictions of data set with n samples by Eq. (25). The complexity for calculating sample importance in (30) is $O(n)$. In the pseudo-labeling step, the unlabeled data are pseudo-labeled with a cost of $O(n-m)$ computational time. The classifier updating, sample importance calculating, and pseudo-labeling steps are repeated multiple times until the algorithm ends. Therefore, the major time complexity of the proposed algorithm scales with $O(p(n^3+n^2)+kn^2)$, where p is the number of loops in Algorithm 1.

3.3 Discussions with related works

There are some works related with the proposed SSPL method. For example, Zhao et al. have proposed a semi-supervised learning method called Learning from Local and Global Discriminative Information (LLGDI) [37]. Comparably, LLGDI is a linear method while SSPL is nonlinear; to character the local manifold structure, LLGDI utilizes local regression model, while SSPL utilizes locally linear reconstruction; to measure the complexity of the learner, LLGDI adopts Frobenius norm of the projection matrix, while SSPL adopts the norm of the classification function in RKHS; LLGDI makes use of local and global regression model to learn, while SSPL makes use of SPL regime to learn a more and more mature model. Besieds, [38] proposed a semi-supervised dimensionality reduction (DR) method called soft label-based linear discriminant analysis (SL-LDA), which performs label propagation to get the predicted soft labels of unlabeled samples and then incorporates the soft labels into LDA. SL-LDA and SSPL are much different, e.g., SL-LDA is a linear DR method, while SSPL is a nonlinear classification method; SL-LDA is a two-stage approach which firstly generates the soft

labels and then learns the projection matrix for DR, while SSPL iterates among classifier updating, sample importance calculating and pseudo-labeling in a self-paced fashion; SL-LDA utilizes LDA to get the projection matrix, while SSPL derives the classification function based on the MR framework. In addition, Zhao et al. [39] proposed a semi-supervised label propagation method named compact graph-based semi-supervised learning (CGSSL) for image annotation. Compared with SSPL which formulates the graph weight by locally linear reconstruction, CGSSL proposes a compact local reconstruction graph with symmetrization and normalization; compared with SSPL which is an inductive learning method that has explicit classification function, CGSSL is an transductive learning method that directly predicts the labels of unlabeled samples; besides, compared with SSPL which adopts a self-paced regime to iterate among classifier updating, sample importance calculating and pseudo-labeling, CGSSL adopts a label propagation strategy that each unlabeled sample receives label from its neighborhoods and its own label.

4 Experiments

In this section, we first introduce the utilized experimental benchmark data sets and the compared algorithms. Then we present the experimental settings and experimental results.

4.1 Data sets and the compared algorithms

We implement experiments on five image data sets to compare the proposed method with other methods. The utilized five data sets include: the MIT CBCL data set,[1] the Altkom and the BANCA data sets,[2] the CMUPIE data set [40],and the ORL data set[3] [41].

The MIT CBCL data set contains 6977 images of two classes: 2429 face images and 4548 non-face images. Each image has 19×19 pixels and is reshaped into a 361-dimensional vector. In this section, we will use the whole data set to do experiments.

The Altkom data set contains 1200 face images of 80 persons, that is, there are 15 images for each person. The BANCA face data set consists of 520 face images of 52 persons, i.e., there are 10 images for each person. All images in Altkom and BANCA data sets are normalized to 46×56 pixels using manually labeled eye positions, and are transformed into a 2576-dimensional vectors.

CMUPIE face database consists of 41,368 images of 68 persons. For each person, the images were taken under different poses, illumination conditions, and expressions. We randomly select 2000 images from the database. Therefore, the experimental CMUPIE data set consists of 2000 images belonging to 68 classes. Each image is resized to have 32×32 pixels and is transformed into a 1024-dimensional vector.

The ORL face database consists of 400 images of 40 persons, i.e., there are 10 images for each person. The images were taken for each person at different times, varying the lighting, facial expressions (open or closed eyes, smiling or not smiling), and facial details (glasses or no glasses). The images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). Each image from this database is resized to 32×32 pixels and reshaped to be a 1024-dimensional vector.

Besides, to alleviate the negative effect caused by the different scales of different dimensions, for all the data sets, each row of the training data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ is normalized to make the maximum component of the row being 1.

The following SSC methods are compared in the experiments: the proposed SSPL, the LapRLSC method [15], the GRF method [12, 13], the linear neighborhood propagation (LNP) method [42] and the 1-NN classifier, where the 1-NN classifier is used as the baseline. For SSPL, LapRLSC and GRF methods, we make use of the Gaussian kernel, and search the kernel parameter in the range of σ from $\{0.1\sigma_0, 0.3\sigma_0, 0.5\sigma_0, \ldots, 1.3\sigma_0\}$, where σ_0 is the mean of L_2-distance of all the training samples. As to the complexity regularization parameter γ_K and the smoothness regularization parameter γ_I of SSPL and LapRLSC methods, we follow the strategy adopted in [15] which sets $CK = \gamma_K m$, $CI = \frac{\gamma_I m}{n^2}$. We find that the algorithms perform well with a wide range of parameters CK and CI. For convenience, we simply set $CK = 0.005$ and $CI = 0.1$ for all data sets. Besides, the neighborhood size k to build the k-NN graph is empirically set to be 20 for CBCL data set, 7 for CMUPIE and Altkom data sets, and 4 for BANCA and ORL data sets.

4.2 Experimental settings and results

The experiments on the five data sets are implemented with the following settings. We first randomly select 85% images of each data set as the training set $\mathcal{X}_{\mathrm{Tr}}$, and the rest samples as the test set $\mathcal{X}_{\mathrm{Te}}$. Then,for the training set $\mathcal{X}_{\mathrm{Tr}}$, under a specific number of labeled training points (m), we carry out tenfold cross validation (10-CV) to generate the validation set and the labeled points of training set, with the

[1] http://cbcl.mit.edu/software-datasets.

[2] http://www.iis.ee.ic.ac.uk/icvl/code.htm.

[3] http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html.

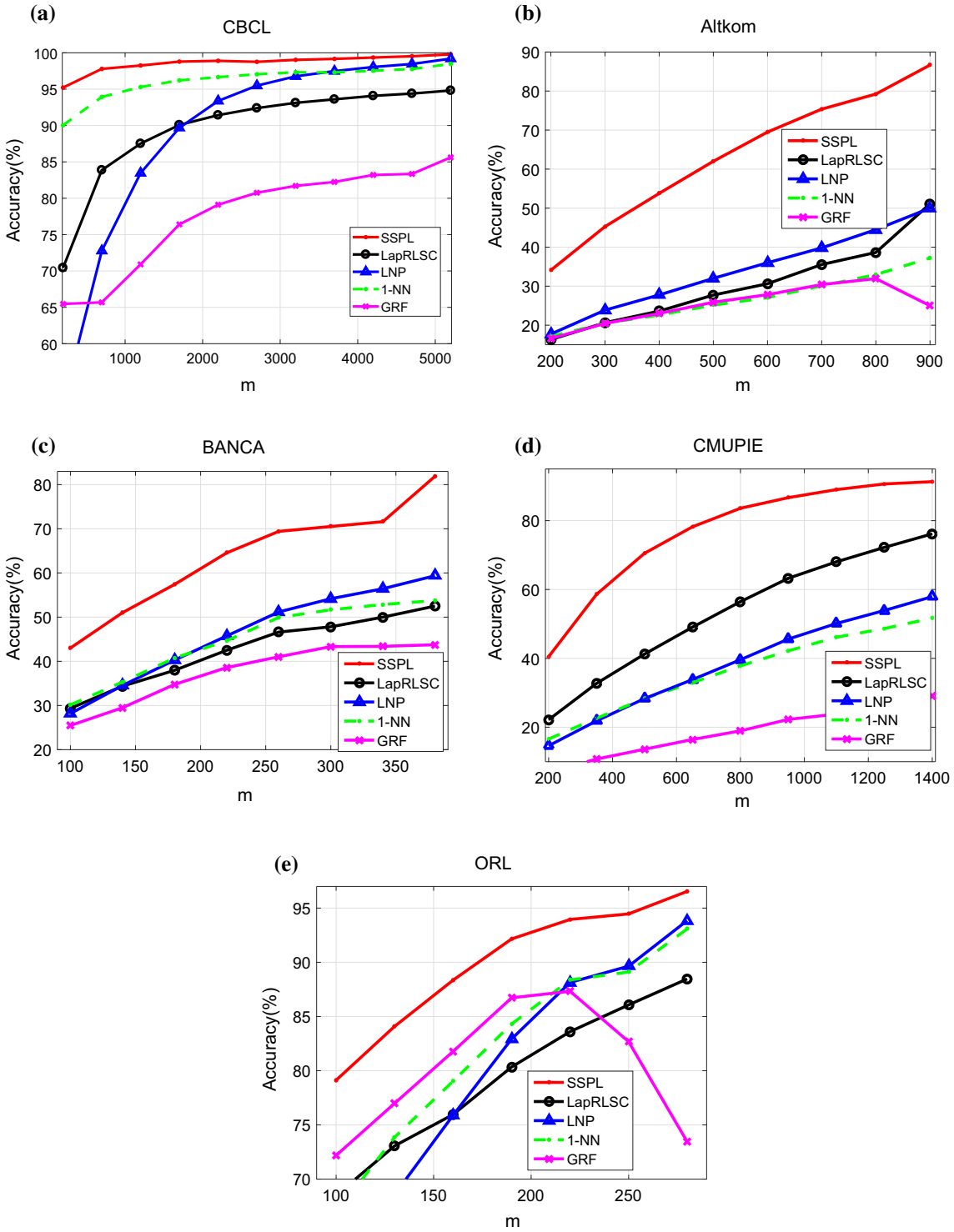

Fig. 1 Classification results on the unlabeled samples of the training sets, where x-axis represents the number of labeled data points in the training set, and y-axis represents the corresponding classification accuracy

rest images forming the unlabeled points of training set. For the proposed SSPL method, the age parameter λ and the kernel parameter σ are tuned on this validation set. For LapRLSC and GRF methods, the kernel parameter σ is also tuned on this validation set, to make a fair comparison. Once we have learned the classifiers, classifications are performed on the unlabeled points in the training set, and on the test set $\mathcal{X}_{\mathrm{Te}}$,respectively.

Table 1 Classification results (accuracy rates %, and p values in the parentheses) on the test sets with the number of labeled points (m) varying

Data	m	SSPL	LapRLSC	LNP	1-NN	GRF
CBCL	200	96.26	69.28 (0.0000)	79.35 (0.0000)	90.72 (0.0000)	63.32 (0.0000)
	1200	98.62	90.71 (0.0000)	96.43 (0.0000)	95.85 (0.0000)	66.07 (0.0000)
	2200	98.96	95.12 (0.0000)	97.60 (0.0000)	96.94 (0.0000)	76.39 (0.0000)
	3200	99.08	97.28 (0.0001)	98.23 (0.0000)	97.57 (0.0000)	80.65 (0.0000)
Altkom	400	61.06	48.33 (0.0000)	30.83 (0.0000)	25.50 (0.0000)	23.67 (0.0000)
	500	70.06	57.44 (0.0000)	34.67 (0.0000)	27.06 (0.0000)	25.78 (0.0000)
	600	77.72	64.94 (0.0000)	39.17 (0.0000)	29.22 (0.0000)	28.44 (0.0000)
	700	82.11	72.22 (0.0000)	41.89 (0.0000)	31.78 (0.0000)	30.67 (0.0000)
BANCA	180	64.63	59.63 (0.0002)	46.00 (0.0000)	42.75 (0.0000)	37.00 (0.0000)
	220	72.13	65.88 (0.0011)	50.25 (0.0000)	47.13 (0.0000)	40.75 (0.0000)
	260	77.50	71.75 (0.0006)	54.13 (0.0000)	50.00 (0.0000)	43.63 (0.0000)
	300	80.75	75.50 (0.0002)	57.63 (0.0000)	54.00 (0.0000)	47.75 (0.0000)
CMUPIE	350	56.8	39.87 (0.0000)	26.23 (0.0000)	22.00 (0.0000)	8.57 (0.0000)
	500	69.60	51.80 (0.0000)	33.07 (0.0000)	28.13 (0.0000)	11.30 (0.0000)
	650	77.53	60.60 (0.0000)	38.17 (0.0000)	32.73 (0.0000)	13.80 (0.0000)
	800	81.73	68.23 (0.0000)	42.47 (0.0000)	36.47 (0.0000)	16.70 (0.0000)
ORL	100	82.67	71.50 (0.0001)	71.33 (0.0001)	69.00 (0.0000)	76.83 (0.0064)
	130	88.00	78.50 (0.0000)	76.83 (0.0000)	75.67 (0.0000)	80.50 (0.0006)
	160	92.33	85.50 (0.0000)	82.83 (0.0000)	81.50 (0.0000)	85.50 (0.0013)
	190	95.33	89.67 (0.0001)	85.50 (0.0006)	85.00 (0.0002)	88.17 (0.0020)

For the five data sets, the classification accuracies of the compared algorithms for the unlabeled samples in the training sets are shown in Fig. 1, with the number of labeled data points (m) changing. From the results we can see that, the proposed SSPL method performs best among the compared methods. Though both based on the manifold regularization framework, the proposed SSPL method outperforms the LapRLSC method. There may be several reasons. First, compared with LapRLSC method, SSPL learns the model in a self-paced way that initially trains a rough model on easy samples, and then gradually incorporates more and more complex samples, to train a mature model. Second, LapRLSC makes use of Gaussian kernel to build the relationship between neighborhood points, while SSPL utilizes locally linear reconstruction which could be more adaptive and flexible to model the relationship between neighborhood points. Finally, SSPL takes account of data importance in the training process, while LapRLSC treats all training samples as equal importance. Besides, compared with LNP algorithm which propagates the labels from the labeled points to the whole data set using the locally linear neighborhoods with sufficient smoothness, the proposed SSPL method not only considers the smoothness of the classification function, but also takes account of the complexity of the classification function in the reproducing kernel Hilbert space, and meanwhile, the proposed SSPL adopts the self-paced regime which can help to extract reliable knowledge from training samples.

The classification results of the algorithms on the test sets are shown in Table 1 for several representative values of m, where the best classification results are in boldface for each specific value of m. The value in parenthesis is the p value of the paired t-test between the proposed SSPL method and other methods. From the statistical tests, we can see that the discriminative ability of the proposed SSPL is significantly better than other algorithms. Besides, it should be pointed out that, the proposed SSPL method and LapRLSC method are inductive, and the explicit classification function learned by the training samples can be utilized to predict the labels of test data points. Conversely, the rest methods are transductive; therefore, in order to predict the labels of test points, we should run the algorithm again by combing training and test samples, which is time-consuming. In this way, the proposed SSPL obtains both effectiveness and efficiency.

To further analyze whether self-paced learning regime contributes to performance of the proposed SSPL method, we implement a variant of SSPL, named SSPL (w/o SPL), and compare the classification performance of SSPL (w/o SPL) with SSPL. In detail, SSPL (w/o SPL) algorithm can be formulated as:

$$\min_{f_s\in\mathcal{H}_K,\{\mathbf{z}_i\}_{i=m+1}^{n}} \left\{ \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{z}_i-\mathbf{f}(\mathbf{x}_i)\|_2^2+\gamma_K\sum_{s=1}^{C}\|f_s\|_K^2 + \frac{\gamma_I}{n^2}\sum_{i=1}^{n}\|\mathbf{f}(\mathbf{x}_i)-\sum_{p=1}^{n}M_{ip}\mathbf{f}(\mathbf{x}_p)\|_2^2 \right\}. \tag{33}$$

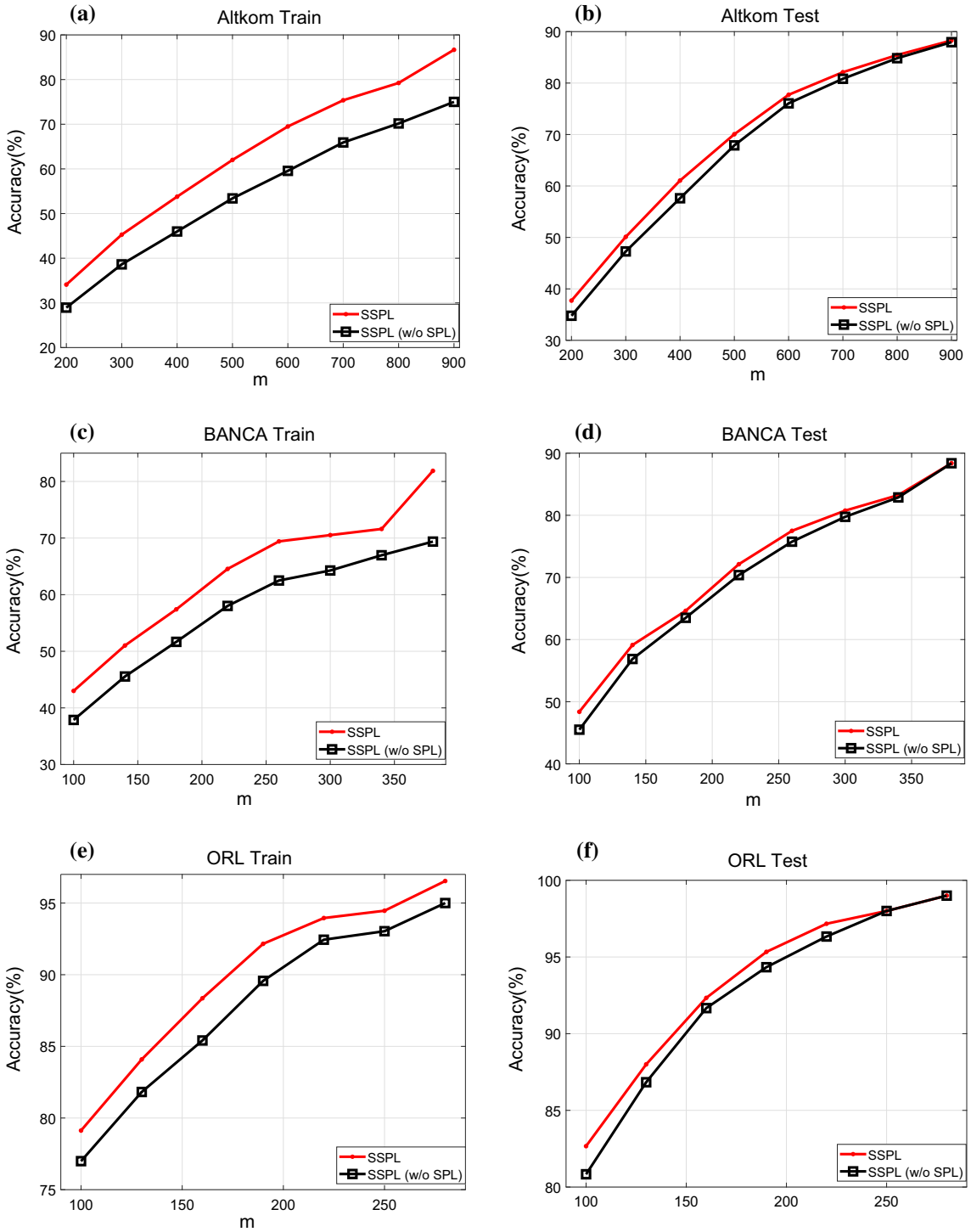

Fig. 2 Classification results of SSPL and SSPL (w/o SPL) on the unlabeled samples of the training sets (left column) and on the test samples (right column), where x-axis represents the number of labeled data points in the training set, and y-axis represents the corresponding classification accuracy

Compared with the SSPL formulation (14), SSPL (w/o SPL) sets the importance values $v_i (i = 1, \ldots, n)$ of all training samples to be 1, i.e., SSPL (w/o SPL) removes the self-paced learning regime of SSPL. In the optimization process of SSPL (w/o SPL), the pseudo-labels $\mathbf{z}_i (i = m+1, \ldots, n)$ of unlabeled samples are initialized as $\mathbf{0}$, and then SSPL (w/o SPL) utilizes AOS to iterate between classifier updating and pseudo-labeling of unlabeled samples. In detail, the classifier can be updated via (24) and

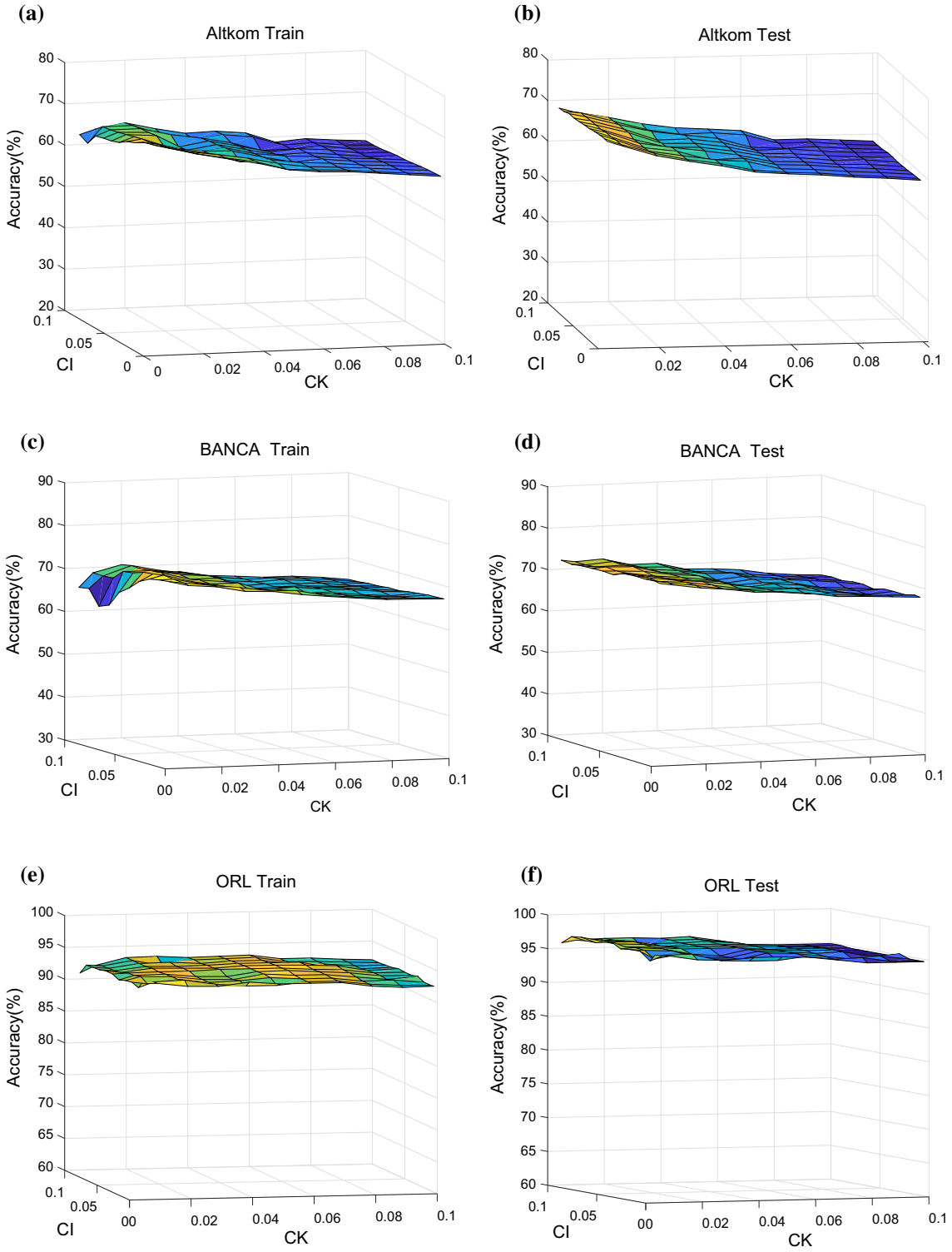

Fig. 3 Classification results on the unlabeled samples of the training sets (left column) and on the test sets (right column), with the regularization parameters *CK* and *CI* varying

(25) with **V** being the identity matrix, and the pseudo-labels of unlabeled samples can be assigned by (32). We compare the performances of SSPL and SSPL (w/o SPL) on Altkom, BANCA and ORL data sets, and the classification accuracies for the unlabeled samples in the training sets and for the test samples are shown in Fig. 2. From the

results we can see that SSPL method outperforms SSPL (w/o SPL), in other words, the self-paced learning regime significantly enhances the classification performance of SSPL.

Besides, to show the sensitiveness of the parameter setting in the proposed SSPL method, we compare the performance of SSPL with different regularization parameters, on Altkom, BANCA and ORL data sets. Here we follow the strategy adopted in the MR framework [15], which utilizes two intermediate parameters $CK = \gamma_K m$, $CI = \frac{\gamma_I m}{n^2}$, where γ_K is the complexity regularization parameter and γ_I is the smoothness regularization parameter, then we change CK and CI in the range $\{0.005, 0.01, 0.02, \ 0.03, \ldots, 0.1\}$. Figure 3 shows the classification accuracies for the unlabeled samples in the training sets and for the test samples. From the figure, we can see that the proposed SSPL performs well with a wide range of parameters CK and CI.

5 Conclusion

In this paper, we proposed a novel semi-supervised classification algorithm called structure regularized self-paced learning (SSPL) method. SSPL integrates self-paced learning paradigm, which learns the model gradually from easy to complex samples, into the manifold regularization framework for semi-supervised learning. The proposed method learns the model by iterating among classifier updating, sample importance calculating and important unlabeled sample pseudo-labeling. With an adaptive pace from easy to hard samples, the learner can extract reliable knowledge from training data, and the labels can be propagated from labeled samples to unlabeled samples. Besides, SSPL defines a new kind of mixture loss which can adaptively determine the sample importance for the subsequent classifier, without need of manually designing. Finally, the proposed method has an explicit multi-class classification function for new samples. Experiments have been conducted on several data sets, and the classification results have shown the recognition superiority of the proposed method. Despite being able to deliver promising results for semi-supervised classification, SSPL can be further improved in the future. For example, instead of solving image classification problem, how to utilize SSPL for some other real-world applications, such as electronic book analysis [43], is another challenge and of great importance.

Acknowledgements This work was supported in part by the National Natural Science Foundation (NNSF) of China [Grant Numbers: 61503263, 61772373, 61772374], in part by the Zhejiang Provincial Natural Science Foundation [Grant Numbers: LY15F030011, LY17F030004], in part by the Project of science and technology plans of Wenzhou City [Grant Number: G20160002].

References

1. Gong C, Tao DC, Maybank SJ, Liu W, Kang GL, Yang J (2016) Multi-modal curriculum learning for semi-supervised image classification. IEEE Trans Image Process 25(7):3249–3260
2. Liu CL, Hsaio WH, Lee CH, Chang TS, Kuo TS (2016) Semi-supervised text classification with universum learning. IEEE Trans Cybern 46(2):462–473
3. Huang H, Feng HL (2012) Gene classification using parameter-free semi-supervised manifold learning. IEEE/ACM Trans Comput Biol Bioinform 9(3):818–827
4. Reitmaier T, Calma A, Sick B (2015) Transductive active learning—a new semi-supervised learning approach based on iteratively refined generative models to capture structure in data. Inform Sci 293:275–298
5. Fujino A, Ueda N, Saito K (2008) Semi-supervised learning for a hybrid generative/discriminative classifier based on the maximum entropy principle. IEEE Trans Pattern Anal Mach Intell 30(3):424–437
6. Maulik U, Chakraborty D (2011) A self-trained ensemble with semisupervised SVM: an application to pixel classification of remote sensing imagery. Pattern Recogn 44(3):615–623
7. Wu D, Shang MS, Luo X, Xu J, Yan HY, Deng WH, Wang GY (2017) Self-training semi-supervised classification based on density peaks of data. Neurocomputing. https://doi.org/10.1016/j.neucom.2017.05.072
8. Li M, Zhou ZH (2007) Learning techniques using undiagnosed samples. IEEE Trans Syst Man Cybern Part A 37(6):1088–1098
9. Xu YK, Qin L, Huang QM (2016) Coupling reranking and structured output SVM co-train for multitarget tracking. IEEE Trans Circuits Syst Video Technol 26(6):1084–1098
10. Chapelle O, Sindhwani V, Keerthi SS (2008) Optimization techniques for semi-supervised support vector machines. J Mach Learn Res 9:203–233
11. Lu ZW, Wang LW (2015) Noise-robust semi-supervised learning via fast sparse coding. Pattern Recogn 48(2):605–612
12. Zhu X, Ghahramani Z, Lafferty J (2003) Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of the 20th international conference on machine learning (ICML2003)
13. Zhu X, Ghahramani Z (2002) Learning from labeled and unlabeled data with label propagation. Technical Report CMUCALD-02-107, Computer Science Department, Carnegie Mellon University
14. Zhou D, Bousquet O, Lal T, Weston J, Schökopf B (2014) Learning with local and global consistency. In: Proceedings of the neural information processing systems conference (NIPS 2004)
15. Belkin M, Sindhwani V, Niyogi P (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 7:2399–2434
16. Zhao MY, Jiao LC, Feng J, Liu TY (2014) A simplified low rank and sparse graph for semi-supervised learning. Neurocomputing 140:84–96
17. Zhuang LS, Zhou ZH, Gao SH, Yin JW, Lin ZC, Ma Y (2017) Label information guided graph construction for semi-supervised learning. IEEE Trans Image Process 26(9):4182–4192
18. Chapelle O, Weston J, Schökopf B (2003) Cluster kernels for semisupervised learning. In: Proceedings of the neural information processing systems conference (NIPS2003), pp 585–592

19. Chapelle O, Scholkopf B, Zien A (2006) Semi-supervised learning. MIT Press, Cambridge
20. Wang YY, Chen SC, Zhou ZH (2012) New semi-supervised classification method based on modified cluster assumption. IEEE Trans Neural Netw 23(5):689–702
21. Zhu X (2006) Semi-supervised learning literature survey. Technical Report 1530, Computer Science Department, University of Wisconsin
22. Kumar M, Packer B, Koller D (2010) Self-paced learning for latent variable models. In: Proceedings of the neural information processing systems conference (NIPS2010), pp 1189–1197
23. Meng DY, Zhao Q, Jiang L (2017) A theoretical understanding of self-paced learning. Inform Sci 414:319–328
24. Jiang L, Meng DY, Yu SI, Lan ZZ, Shan SG, Hauptmann A (2014) Self-paced learning with diversity. In: Proceedings of the neural information processing systems conference (NIPS2014)
25. Zhang DW, Meng DY, Han JW (2017) Co-saliency detection via a self-paced multiple-instance learning framework. IEEE Trans Pattern Anal Mach Intell 39(5):865–878
26. Lin L, Wang KZ, Meng DY, Zuo WM, Zhang L (2017) Active self-paced learning for cost-effective and progressive face identification. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/TPAMI.2017.2652459
27. Supančič III J, Ramanan D (2013) Self-paced learning for long-term tracking. In: IEEE conference on computer vision and pattern recognition (CVPR2013), pp 1189–1197
28. Kumar M, Turki H, Preston D, Koller D (2011) Learning specific-class segmentation from diverse data. In: IEEE conference on computer vision and pattern recognition (CVPR2011), pp 1800–1807
29. Yu S et al (2014) Cmu-informedia@ trecvid 2014 multimedia event detection. In: TRECVID video retrieval evaluation workshop
30. Bengio Y, Louradour J, Collobert R, Weston J (2009) Curriculum learning. In: Proceedings of the 20th international conference on machine learning (ICML2009)
31. Jiang L, Meng D, Mitamura T, Hauptmann A (2014) Easy samples first: self-paced reranking for zeroexample multimedia search. In: Proceedings of ACM multimedia
32. Zhao Q, Meng DY, Jiang L, Xie Q, Xu ZB, Hauptmann A (2015) Self-paced learning for matrix factorization. In: Proceedings of AAAI conference on artificial intelligence (AAAI2015)
33. Bazaraa M, Sherali H, Shetty C (1993) Nonlinear programming—theory and algorithms. Wiley, New York
34. Jiang L, Meng DY, Zhao Q, Shan SG, Hauptmann A (2015) Self-paced curriculum learning. In: Proceedings of AAAI conference on artificial intelligence (AAAI2015)
35. Aronszajn N (1950) Theory of reproducing kernels. Trans Am Math Soc 68(3):337–404
36. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290:2323–2326
37. Zhao MB, Chow Tommy WS, Wu Z, Zhang Z, Li B (2015) Learning from normalized local and global discriminative information for semi-supervised regression and dimensionality reduction. Inform Sci 324:286–309
38. Zhao MB, Zhang Z, Chow Tommy WS, Li B (2014) A general soft label based Linear Discriminant Analysis for semi-supervised dimensionality reduction. Neural Netw 55:83–97
39. Zhao MB, Chow Tommy WS, Zhang Z, Li B (2015) Automatic image annotation via compact graph based semi-supervised learning. Knowl based Syst 76:148–165
40. Gross R, Baker S, Matthews I (2005) Generic vs. person specific active appearance models. Image Vis Comput 23(11):1080–1093
41. Samaria FS, Harter AC (1994) Parameterisation of a stochastic model for human face identification. In: Proceedings of the second IEEE workshop on applications of computer vision, pp 138–142
42. Wang F, Zhang CS (2008) Label propagation through linear neighborhoods. IEEE Trans Knowl Data Eng 20(1):55–67
43. Zhang HJ, Chow Tommy WS, JonathanWu QM (2016) Organizing books and authors by multilayer SOM. IEEE Trans Neural Netw 27(12):2537–2550