Pattern Recognition 45 (2012) 2227–2236

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Simultaneous clustering and classification over cluster structure representation

Qiang Qian [a], Songcan Chen [a,*], Weiling Cai [b]

[a] Department of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, PR China
[b] Department of Computer Science and Technology, Nanjing Normal University, Nanjing 210097, PR China

ARTICLE INFO

Article history:
Received 24 February 2011
Received in revised form
27 October 2011
Accepted 18 November 2011
Available online 13 December 2011

Keywords:
Structure in data
Clustering learning
Classification learning
Simultaneous classification and clustering learning

ABSTRACT

Two main tasks in pattern recognition area are clustering and classification. Owing to their different goals, traditionally these two tasks are treated separately. However, when label information is available, such separate treatment can not fully explore data information. First, classification is not favored by the data cluster structure. Second, clustering is not guided by valuable label information. Third, the relationship of clusters and classes is not revealed. Contrary to this separate learning treatment, simultaneous learning clustering and classification could benefit each other and overcomes these problems.

Recently, a simultaneous learning framework SCC was proposed. Through modeling $p(class|cluster)$ classification and clustering mechanism in SCC depend only on cluster centroids. However, it produces severely nonlinear objective, thus has to use a heuristic searching method, modified Particle Swarm Optimization, to find the optimal solution. But it is very slow. Further, modeling $p(class|cluster)$ makes SCC hard to incorporate semi-supervised settings.

In this paper, we propose an alternative framework SC^{3}SR for simultaneous learning. Besides a classifier derived on the original data, another classifier on the newly-formed cluster structure representation is derived as well. Through this classifier, the clustering learning is guided by the label and classification learning is also favored by cluster structure of data. The final objective is continuously differentiable for which some principled optimization algorithms with convergence guaranteed exist. As a result, our algorithm is much faster than SCC. Further, we generalize this framework to semisupervised situation with the idea of manifold regularization and propose SemiSC^{3}SR algorithm. Our experiments demonstrate the effectiveness of both SC^{3}SR and SemiSC^{3}SR.

1. Introduction

In traditional pattern recognition area, two main tasks are clustering and classification [11]. Clustering task is unsupervised in general. It groups similar instances into several meaningful clusters, and is usually used to find out data intrinsic structure. Classification task is usually supervised. It first learns a classifier from training data with labels, and uses the learned classifier to attach suitable labels to testing data. These two tasks are usually treated separately, because both goals are quite different. However, when label information is available, simultaneous learning clustering and classification tasks will benefit from each other [5].

First, classification task will be favored when data intrinsic structure is considered. In pattern recognition area, well-known No Free Lunch theorem tells that no classifier is innately superior to any other classifier unless it incorporates the prior knowledge [9]. For classification, two kinds of prior knowledge are usually considered. One is on classification function, for example, the function is usually assumed to be smooth to avoid overfitting [10,24]. The other is on data distribution, of which there are two types of popular assumptions, cluster and manifold assumptions [7]. Cluster assumption assumes that the points of each class tend to form clusters, and manifold assumption assumes that data lie on a low dimension manifold. This paper focuses on simultaneous learning clustering and classification tasks, hence cluster assumption is used as data prior knowledge. It suggests that when data intrinsic structure revealed by some clustering algorithm is incorporated into the classification task, better performance may be desired [5,13,25].

Second, clustering task will be guided by valuable label information. For example, when clustering faces, the Euclidean distance between the faces from the same person may be larger than the Euclidean distance between the ones from the different persons due to varying lighting [20]. However, one may want to

* Corresponding author. Tel.: +86 25 84896481x12221; fax: +86 25 84892400.
E-mail addresses: qian.qiang.yx@gmail.com (Q. Qian), s.chen@nuaa.edu.cn (S. Chen), caiwl@nuaa.edu.cn (W. Cai).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.11.027

put the faces from the same person into his(her) own clusters and push the faces from the different persons away even they are under similar lighting condition. When label information is available, the faces from the different persons will be forcibly pushed apart according to the guidance of label information, consequently faces from the same persons and under the same lighting condition will be grouped together.

Third, the relation between clusters and classes can be revealed, so some meaningful insight between clusters and classes could be caught. For example, we can know whether the cluster contains single class data or not, and by which clusters the class is formed.

Recently, a simultaneous learning framework for clustering and classification SCC was proposed [5]. In order to bridge the clustering and classification tasks, SCC proposed to model *cluster posterior probabilities of classes* $p(class|cluster)$. Then by assuming that class is conditionally independent with data given cluster, the classification mechanism was broken down to $p(class|instance) = \sum p(class|cluster)p(cluster|instance)$ according to Bayesian formula. As a result the classification and clustering tasks are naturally and seamlessly fused together, and the relation between class and cluster is revealed explicitly by $p(class|cluster)$. In its implementation, SCC estimates $p(class|cluster)$ and $p(cluster|instance)$ only by cluster centroids and designs the objective with cluster centroids be sole argument. However, it is highly nonlinear [5] so hard to find the local optimal solution. In fact, SCC has to resort to a heuristic searching method, modified Particle Swarm Optimization method, to find the local optimal solution. But it is slow, sensitive to initial settings and has no guarantee of convergence. Furthermore, SCC is not capable of semi-supervised situation, because for unlabeled data, $p(class|cluster)$ cannot be estimated [5].

In this paper, we propose an alternative framework called SC^{3}SR (Simultaneous Clustering and Classification over Cluster Structure Representation) for simultaneous learning. The key point connecting clustering and classification tasks is classifier designed on cluster structure representation (CSR) rather than *cluster posterior probabilities of classes* $p(class|cluster)$. Actually, besides the classifier designed on the original data, another classifier on its CSR is also jointly designed. For their consistence, their disagreement is minimized. Consequently, the classifier designed on the original data is influenced by the data cluster structure, while data clustering procedure is also guided by label information. The final objective is continuously differentiable with blockwise arguments, thus can be optimized through block coordinate descent algorithm which guarantees that the iteration procedure converges to a stationary point [2]. Experiments demonstrate that our algorithm is much faster than SCC. The difference with SCC is that our algorithm can implicitly rather than explicitly give out the relation between cluster and class $p(class|cluster)$. But our framework is more flexible. Since SC^{3}SR derives two classifiers from both the original data and the CSR, it could be easily generalized to deal with semi-supervised situation with the idea of manifold regularization [1]. Consequently we also develop incidentally SemiSC^{3}SR algorithm in this paper and preliminarily examine its effectiveness by comparing with some popular semi-supervised algorithms. Furthermore, our framework can also be viewed as a dimension reduction method since the learned CSR can be also taken as a reduced-dimensionality representation of the original data. Finally, we enumerate the differences between SCC and SC^{3}SR in Table 1 and give out our contributions as follows:

- Propose a new simultaneously learning classification and clustering algorithm SC^{3}SR which is much faster and more flexible than SCC.
- Generalize SC^{3}SR to semi-supervised situation and develop SemiSC^{3}SR algorithm.
- Examine their effectiveness and efficiency through experiments.

Table 1
Difference between SCC and SC^{3}SR.

Different points	SCC	SC^{3}SR
Objective dependence	Cluster centroid	CSR
Objective function	Hard to optimize	Easy to optimize
Theoretical convergence guarantee	No	Yes
$p(class\|cluster)$	Explicit	Implicit
Semi-supervised situation	No	Yes
Dimension reduction	No	Yes

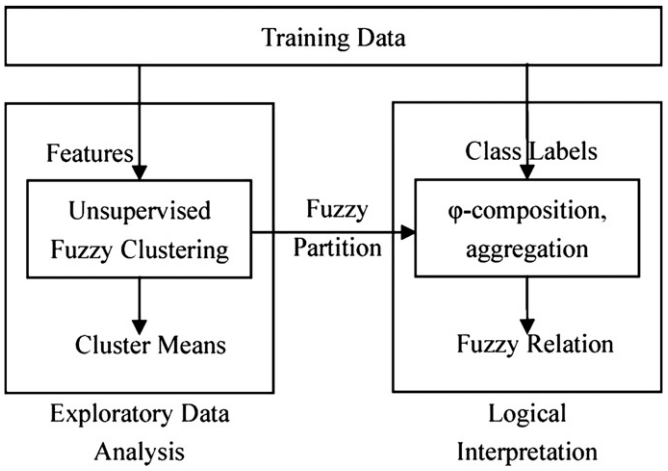


Fig. 1. Training of FRC and RFRC [22].

The rest of paper is organized as follows: in Section 2, some related works are reviewed. In Section 3, Both SC^{3}SR and SemiSC^{3}SR are proposed. Experimental results are presented in Section 4. And Section 5 concludes this paper.

2. Related work

Much effort has been devoted to connecting clustering and classification tasks together, most of which treat clustering and classification in a sequential manner.

In popular 3-layer radial basis function neural networks (RBFNN) [18,25], clustering algorithms like *k*-means and fuzzy *c*-means (FCM) are first performed to help to determine the parameters of hidden layer, next the connection weights between the hidden layer and output layer are optimized by minimizing the mean square error between the target and actual outputs. Better generalization is obtained through incorporating clustering information [5]. However, clustering task does not benefit from label information [5].

In fuzzy relational classifier (FRC) algorithm proposed by Setnes et al. [22], the training data are first clustered by FCM, then a fuzzy relation between clusters and given classes is artificially established explicitly by means of a fuzzy composite operator. The whole procedure is illustrated in Fig. 1 [22]. Lately, Cai et al. [4] presented robust FRC(RFRC) algorithm to enhance the robustness of FRC from two aspects. First, the authors used the robust kernel FCM [8] as the clustering tool; second motivated by fuzzy *k*-nearest-neighbor algorithm [12], they replaced hard class label with the soft class label to construct their fuzzy relation, as a result significantly boosting the robustness and accuracy of FRC. However, a common disadvantage in FRC and RFRC is that the

fuzzy relation between clusters and classes is hard to optimize due to the indifferentiability and complexity of the composite operators and lacks the probabilistic interpretation thus fails to reveal the reliable relation between clusters and classes.

Unlike the RBFNN, FRC and RFRC algorithms which conduct clustering task in unsupervised manner, some algorithms take supervised clustering algorithm to aid the classification tasks. VQ+LVQ3 algorithm proposed by Kim et al. [13] utilizes clustering method to reduce the computation burden in nearest neighbor classifier while without sacrificing the classification performance. In the clustering procedure, LVQ3 algorithm which employs the label information is used to determine the cluster center and cluster labels. Similar to VQ+LVQ3, CCAS [26,14] and its extension ECCAS [15] also employ a supervised clustering procedure to find out a set of prototypes. Both VQ+LVQ3 and (E)CCAS algorithms use nearest neighbor classifier in classification phrase, they actually do not have training phrases. The common idea is to find out the best prototypes as the class representatives for the subsequent nearest neighbor classifier.

Recently, Cai et al. [5] proposed a simultaneous learning clustering and classification algorithm named SCC. In SCC, the clustering and classification tasks are learned simultaneously in one framework. Through modeling *cluster posterior probabilities of classes* $p(class|cluster)$, SCC succeeds in acquiring robust classification and clustering simultaneously and in revealing the underlying relation between clusters and classes. However, the objective of SCC is hard to optimize, thus SCC has to resort to a heuristic searching method, modified Particle Swarm Optimization method, to find the local optimal solution. Further, SCC cannot deal with semi-supervised situation because $p(class|cluster)$ could not be estimated without labels.

The formulation of SCC combines the classification and clustering criteria into a single objective with a trade-off parameter, thus will make compromise between the clustering performance for the classification performance. Later Cai et al. [6] remedied this problem by optimizing the classification and clustering criteria together through multi-objective optimization method rather than the combined single one. However the algorithm, called MSCC, still inherits some flaws of SCC, despite achieving better performance. For example, it is hard to optimize thus has to resort to multi-objective particle swarm optimization method, and can not deal with semi-supervised situation. In this paper, we propose a new formulation to overcome these flaws. Our framework takes the single objective like SCC, thus comparing with MSCC is unfair to our framework because our framework does not take the more effective multi-objective framework. Yet, it can also be generalized into the multi-objective one, but this work is out of our paper's scope.

3. SC^{3}SR: simultaneous clustering and classification over cluster structure representation

In this section, we present our SC^{3}SR algorithm. First, we introduce some notations. Then we explain what we called CSR in this paper. Next, we introduce the mathematical formulation of SC^{3}SR algorithm and its optimization methods. Finally we introduce the kernel version of SC^{3}SR.

3.1. Notations

In this paper, e is the column vectors with all entries being 1. $\|\cdot\|$ is 2-norm.

Let data set be $X=[x_1,\ldots,x_N]\in\mathcal{R}^{D\times N}$ and its corresponding label set be $Y=\{y_1,\ldots,y_N\}$. For each $y_i\in Y$, $y_i\in\{1,2,\ldots,C\}$ where C is the number of classes.

K is the number of clusters. In Fuzzy c-means algorithm, cluster centroid of kth cluster is denoted by v_k. Matrix $V=[v_1,\ldots,v_K]$ contains all the cluster centroids. The cluster indicator vector of ith instance is denoted by h_i with each entry being non-negative and $e^Th_i=1$. Cluster indicator matrix $H=[h_1,\ldots,h_N]$ contains all the cluster indicator vectors.

3.2. CSR

A representation of data is a way describing the data. For example, web pages in the Internet can be represented by the contained words; also they can be represented by the hyperlink graph. CSR is a representation which describes data from cluster perspective and reveals the data cluster structure. Actually, the results of any clustering algorithms could be considered as a CSR since the cluster indicator vectors reveal the data cluster structure. But they are not all. For example, in spectral clustering algorithm, data are first embedded into an Euclidean space according their pairwise similarity relation. Distances between coordinates in the new Euclidean space reflect their similarity relation. Similar instances gather together and dissimilar instances separate. So these coordinates also reveal the data cluster structure and can be considered as a CSR. Here we list some CSRs yielded from some common clustering algorithm.

1. In FCM (k-means), the cluster indicator vectors can be considered as CSR.
2. In spectral clustering, the coordinates in the embedded Euclidean space can be considered a CSR.
3. In pLSA, $p(topic|document)$ can be considered as CSR.

3.3. SC^{3}SR

In simultaneous learning clustering and classification framework, the clustering and classification tasks should be optimized together to benefit from each other. So the key point is how to bridge the clustering and classification tasks. SCC proposed to model *cluster posterior probabilities of classes* $p(class|cluster)$, and bridge the clustering and classification tasks through Bayesian formula. However, it leads to a severely nonlinear optimization problem. In our framework, we bridge the clustering and classification tasks through designing a classifier on the CSR. Actually we design two classifiers. One is on the original data, and the other is on CSR. Since the label is the same, their prediction should be consensus and their disagreement should be minimized.

Based on the above description, the framework is formulated as follows:

$$\min J = C(X,H)+\lambda_1\sum_i(l_{orig}(y_i,f_{orig}(x_i))+l_{csr}(y_i,f_{csr}(x_i))) +\lambda_2\sum_i D(f_{orig}(x_i),f_{csr}(x_i)) \quad (1)$$

The first term measures the cluster cost. X is data matrix and H is the cluster indicator matrix which here is also CSR derived from the clustering algorithm. The smaller the cluster cost is, the better the cluster result is. The second term measures the classification loss of two classifiers, where f_{orig} and f_{csr} are the classifiers designed on original data and the CSR respectively. l_{orig} and l_{csr} are the loss functions measuring the loss of classifier f_{orig} and f_{csr} respectively. The third term $D(\cdot,\cdot)$ measures the output disagreement between classifier f_{orig} and f_{csr}.

This is a general framework. Many clustering algorithms and classification algorithms could be incorporated into this framework. Without loss of generality, we choose FCM as our clustering algorithm and multi-class logistic regression as our classification algorithm.

Since the prediction of multi-class logistic regression is probabilistic, we choose Jensen–Shannon divergence (a Symmetrised divergence of Kullback-Leibler divergence) to measure the output disagreement between f_{orig} and f_{csr}. So the framework is rewritten as follows:

$$\begin{aligned}\min J(V,H,W^x,W^h) &= \sum_i\sum_k h_{ik}^2\|x_i-v_k\|^2 \\ &-\lambda_1\sum_i(\log p(y_i|x_i)+\log p(y_i|h_i)) \\ &+\lambda_2\sum_i\left(\frac{1}{2}KL(p(c|x_i)\|p(c|h_i))+\frac{1}{2}KL(p(c|h_i)\|p(c|x_i))\right) \\ &\text{s.t. } e^Th_i=1 \quad \text{for each } i\in\{1,\ldots,N\} \\ &\quad h_{ik}\geq 0 \quad \text{for each } i\in\{1,\ldots,N\}, \text{ and } k\in\{1,\ldots,K\}\end{aligned} \tag{2}$$

where

$$p(c=1|x_i)=\frac{\exp(x_i^Tw_1^x)}{1+\exp(x_i^Tw_1^x)+\cdots+\exp(x_i^Tw_{C-1}^x)}$$

$$p(c=2|x_i)=\frac{\exp(x_i^Tw_2^x)}{1+\exp(x_i^Tw_1^x)+\cdots+\exp(x_i^Tw_{C-1}^x)}$$

$$\vdots$$

$$p(c=C|x_i)=\frac{1}{1+\exp(x_i^Tw_1^x)+\cdots+\exp(x_i^Tw_{C-1}^x)} \tag{3}$$

and

$$p(c=1|h_i)=\frac{\exp(h_i^Tw_1^h)}{1+\exp(h_i^Tw_1^h)+\cdots+\exp(h_i^Tw_{C-1}^h)}$$

$$p(c=2|h_i)=\frac{\exp(h_i^Tw_2^h)}{1+\exp(h_i^Tw_1^)+\cdots+\exp(h_i^Tw_{C-1}^h)}$$

$$\vdots$$

$$p(c=C|h_i)=\frac{1}{1+\exp(h_i^Tw_1^h)+\cdots+\exp(h_i^Tw_{C-1}^h)} \tag{4}$$

The two constrains over cluster indicator matrix H come from FCM algorithm where the cluster indicator vector in H should be non-negative and sum to one. The arguments are obviously blockwise thus can be optimized with block coordinate descent algorithm which guarantees that the iteration procedure converges to a stationary point of objective function [2].

SC^{3}SR does not produce the probabilistic relation of classes and clusters $p(class_i|cluster_j)$ explicitly in Eq. (2), but by referring to Cai et al. [5], it could be given out after the class labels and cluster labels of data are known. Through Bayesian formula $p(class|cluster)$ can be rewritten as follows:

$$\begin{aligned}p(class_i|cluster_j)&=\frac{p(class_i,cluster_j)}{p(cluster_j)} \\ &=\frac{\#\{x\in class_i, x\in cluster_j\}}{\#\{x\in cluster_j\}}\end{aligned} \tag{5}$$

Clearly this relation is probabilistic because $\sum_i p(class_i|cluster_j)=1$.

3.4. Optimization

The arguments of SC^{3}SR's objective $J(V,H,W^x,W^h)$ are blockwise, thus the objective could be optimized by classic block coordinate descent method. The total arguments in J consists of four blocks V,H,W^x,W^h. Given H,W^x,W^h fixed, V could be optimized by analytically finding out its stationary point. H could be optimized by gradient projection methods since it is constrained. For both W^x and W^h, we use Newton–Raphson method which is commonly used in optimizing logistic regression classifier [3]. The whole algorithm is listed in Algorithm 1.

Algorithm 1. SC^{3}SR.

Input: data matrix X, label vector Y, number of cluster K, maximum iteration number MaxIter
initialize W^h,V,H
while $iter < MaxIter$ **do**
  step 1: update W_x by $W_x = W_x - \alpha_{W_x}H_{W_x}\backslash g_{W_x}$
  step 2: update W^h by $W_h = W_h - \alpha_{W_h}H_{W_h}\backslash g_{W_h}$
  step 3: update each v_k by $v_k=\frac{\sum_i h_{ik}^2x_i}{\sum_i h_{ik}^2}$
  step 4: update each h_i by $h_i = Proj(h_i-\alpha_h g_{h_i})$
end while
calculate $p(class|cluster)$ by Eq. (5).

where g_{W_x},g_{W_h},g_{h_i} are the gradients and H_{W_x},H_{W_h} are Hessian matrices, and the formula of them are given in the Appendix. *Proj* is an operation that projects a vector to the probabilistic simplex. According to [2], for continuously differentiable objective functions, the sequence generated by the block coordinate descent method converges to a stationary point.

3.4.1. Projection onto probability simplex

Each column vector of H is constrained in a probability simplex, so after updating h_i by the gradient descent, the newly-updated point should be projected onto its feasible region, which can be cast as a quadratic programming(QP) problem as follows:

$$\begin{aligned}\min \quad & \tfrac{1}{2}\|z-h_i\|^2 \\ \text{s.t.} \quad & e^Tz=1 \\ & z_k\geq 0 \quad \text{for each } k\in\{1,\ldots,K\}\end{aligned} \tag{6}$$

It is time-consuming to solve the problem by some standard QP packages. Luckily, for such probability simplex constraints, this QP problem can be easily solved [17]. For this problem, we define its Lagrangian problem as follows:

$$\min_{\mu\geq 0} L(\lambda,\mu)=\frac{1}{2}\|z-h_i\|^2+\lambda(e^Tz-1)-\sum_k\mu_kz_k$$

where λ,μ are the Lagrangian multipliers of corresponding constraints. Through its KKT system, we obtain Eq. (7) with respect to the λ, and finally obtain its solution $\lambda^\star$ by the bisection method. Next, the projection $z^\star$ of h_i could be recovered from the $\lambda^\star$ by Eq. (8) which is also derived from KKT system. For detailed information, please refer to Liu et al.'s paper [17].

The whole procedure consists of two steps:

1. solving the equation
$$\sum_k\max(h_{ik}-\lambda,0)-1=0 \tag{7}$$
and finding the root λ^* by the bisection method.
2. recovering the projection z^* from λ^* by formula
$$z_k^*=\max(h_{ik}-\lambda^*,0) \tag{8}$$

The time complexity is only O(K).

3.4.2. Time complexity of SC^{3}SR

Here we consider the time complexity of Algorithm 1. In step 1, optimizing W^x consists of calculating gradient g_{W^x} and Hessian matrix H_{W^x} of W^x and updating W^x by Newton–Raphson method. The time complexities of calculating g_{W^x} and H_{W^x} are $O(N\times D\times C)$ and

$O(N \times D^2 \times C^2)$ respectively. Since updating W^x involves solving the linear system $H_{W^x} \backslash g_{W^x}$(Matlab notation), it costs $O(D^3 \times C^3)$. Consequently step 1 totally costs $O(N \times D^2 \times C^2)+O(D^3 \times C^3)$. Consider the similar role of W^h with W^x in Eq. (2), the time complexity of step 2 is similar with the one of step 1, only replacing data dimension D with number of clusters K, so it is $O(N \times K^2 \times C^2)+O(K^3 \times C^3)$. Step 3 takes $O(N \times K \times D)$. Step 4 virtually consists of updating H and projection which take $O(N \times K \times C)$ and $O(K)$ respectively. Thus step 4 costs $O(N \times K \times C)$. In summary, the time complexity of algorithm 1
is $O(N \times D^2 \times C^2 \times I)+O(D^3 \times C^3 \times I)+O(N \times K^2 \times C^2 \times I)+O(K^3 \times C^3 \times I)+(O(N \times K \times D \times I)$ where I is the iteration number. Despite the lengthy formula, the complexity is much smaller than SCC's. SCC costs $O(P \times N \times C \times K \times D \times I)$ where P is the particle number. According to Cai et al. [5], P and I are set to quite large numbers, 1000 and 500 respectively, to assure finding good solution. In our experiments, the maximum iteration number is set to 100, and it converges typically within 50. So SC^{3}SR is more efficient than SCC.

3.5. SemiSC^{3}SR

Although our major attention focuses on SC^{3}SR, in this subsection, we give out the formulation of semi-supervised SC^{3}SR.

One of the SCC's defects is that it is hard to cope with partially labeled data. The main reason lies in that both the classification and clustering mechanisms depend on the crucial cluster posterior probabilities of classes $p(class|cluster)$. However for unlabeled data, estimating $p(class|cluster)$ is quite difficult since it need know labels of all instances (belonging to some clusters). As a result, SCC cannot be generalized by some classical semi-supervised ideas like manifold regularization. At the same time, SC^{3}SR aims to overcome the difficulty and derives two classifiers respectively on the original data and the CSR. The two classifiers can then be easily generalized to semi-supervised situation with the similar idea of manifold regularization [1]. To this end, we append the Laplacian regularization to the SC^{3}SR's objective and derive the semi-supervised SC^{3}SR (SemiSC^{3}SR) as follows:

$$\begin{aligned}\min\quad & J(V,H,W^x,W^h)=\sum_{i}^{l+u}\sum_{k}h_{ik}^2\|x_i-v_k\|^2\\ & -\lambda_1\sum_{i}^{l}(\log p(y_i|x_i)+\log p(y_i|h_i))\\ & +\lambda_2\sum_{i}^{l+u}\left(\frac{1}{2}KL(p(c|x_i)\|p(c|h_i))+\frac{1}{2}KL(p(c|h_i)\|p(c|x_i))\right)\\ & +\lambda_3\sum_{ij}^{l+u}w_{ij}(\|\log(p(c|x_i))-\log(p(c|x_j))\|^2\\ & +\|\log(p(c|h_i))-\log(p(c|h_j))\|^2)\\ \text{s.t.}\quad & e^Th_i=1\quad \text{for each } i\in\{1,\ldots,N\}\\ & h_{ik}\geq 0\quad \text{for each } i\in\{1,\ldots,N\} \text{ and } k\in\{1,\ldots,K\}\end{aligned}\tag{9}$$

where w_{ij} is the similarity weight between the ith and the jth instances, $\log(p(c|x_i))$ is a vector $[\log(p(c=1|x_i)),\ldots,\log(p(c=C|x_i))]$ and so is $\log(p(c|h_i))$, and l,u are the numbers of the labeled and the unlabeled instances respectively. Here we follow the custom of Laplacian regularization formulation and use the l_2 norm rather than the Jensen–Shannon divergence to measure the difference between the outputs of the ith and the jth instances. For ease of calculation, we use $\log(p(c|x_i))$ instead of $p(c|x_i)$ directly, since $\log(p(c|x_i))$ could be seen as an approximation of $p(c|x_i)$ via $\log(x)\approx x-1$ in the Laplacian regularization.

Like SC^{3}SR, SemiSC^{3}SR can also be optimized by block coordinate descent method and follows the same procedure as Algorithm 1. We only need to replace the corresponding gradients and Hessian matrices with those of SemiSC^{3}SR's objective, and thus omit the algorithm description here while defer the corresponding formulae of gradients and Hessian matrices to Appendix.

3.6. Kernelized SC^{3}SR and SemiSC^{3}SR

In this section, the kernel versions of SC^{3}SR and SemiSC^{3}S are introduced.

Kernel function $k(\cdot,\cdot)$ implicitly induces a map $\phi_i: x\rightarrow \phi_i(x)$(subscript i of ϕ stands for "implicit"). However it changes the number of logistic regression parameters W^x from $(C-1)D$ to $(C-1)N$ for SC^{3}SR algorithm. Using the Newton–Raphson method, directly to optimize the logistic regression parameters will be troublesome since the inverse of Hessian matrix needs computation when N is large.

Consequently, instead of using this implicit kernel mapping, we here adopt the approximated empirical kernel mapping [23,21]. Usually kernel algorithms perform only in the subspace spanned by $\phi_i(x_1),\ldots,\phi_i(x_N)$ in RKHS. This subspace can be embedded into an Euclidean space while all the geometrical structure can still be preserved. Such embedding is called the "empirical kernel mapping" [21] and defined as follows (subscript e of ϕ stands for "empirical"):

$$\phi_e: x\rightarrow \Lambda^{-\frac{1}{2}}U^T[k(x,x_1),\ldots,k(x,x_N)]^T\tag{10}$$

where $K=U\Lambda U^T$ is the eigen-decomposition of kernel matrix K.

To reduce the parameter number, we truncate the small eigenvalues in Λ and use the approximated empirical kernel mapping

$$\phi_e(x_i)=\Lambda^{1/2}U_d(i,:)^T\tag{11}$$

where d is the number of the remained eigenvalues. The number of parameters is then reduced to $(C-1)d$. As a result, $\phi_e(x_1),\ldots,\phi_e(x_N)$ can be directly fed into SC^{3}SR and SemiSC^{3}SR algorithm without any modification.

4. Experiment

In this section, we examine the effectiveness of SC^{3}SR on both clustering and classification tasks to illustrate the merit of simultaneous learning clustering and classification tasks.

4.1. Clustering learning on toy problem

This experiment aims to show the clustering results of SC^{3}SR. As competitors, the results of SCC and FCM are also illustrated. Both kernelized and unkernelized versions are examined. The algorithm names appended by 1 or 2 denote the unkernelized or kernelized versions respectively. RBF kernel is used here. A two-class synthetic dataset which consists of four Gaussian components is generated. The means, covariances and labels are listed in Table 2. The clustering results are demonstrated in Fig. 2. Here as demonstrated in Fig. 2(a), (b), (e), and (f), the simultaneous learning algorithms SC^{3}SR and SCC yield reasonable clustering results respectively. The stretched upper two Gaussian components are not mixed up thanks to the label information guidance.

Table 2
Distribution of synthetic dataset.

Group	Label	Mean	Covariance
1st Gaussian component	1	[−5 −5]	diag([1 1])
2nd Gaussian component	1	[−1 5]	diag([.1 1.5])
3rd Gaussian component	2	[5 −5]	diag([1 1])
4th Gaussian component	2	[1 5]	diag([.1 1.5])

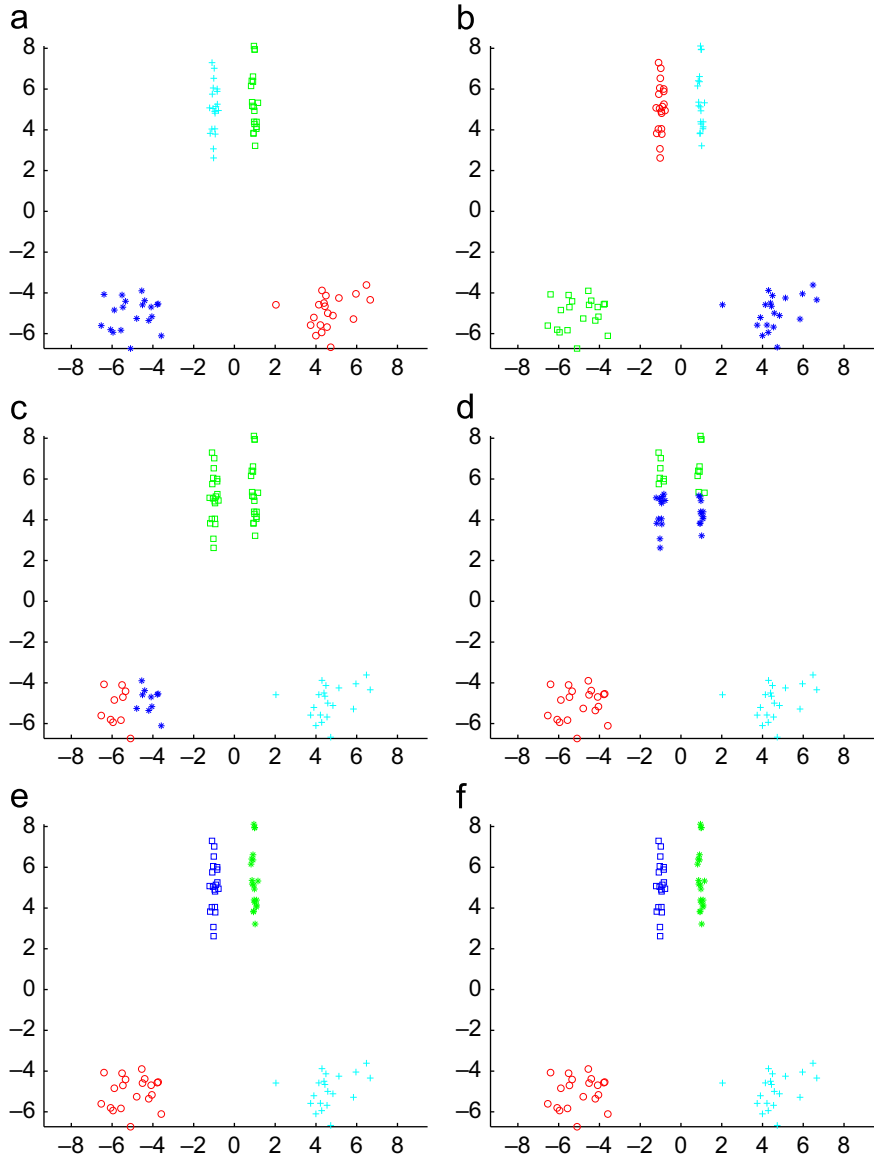


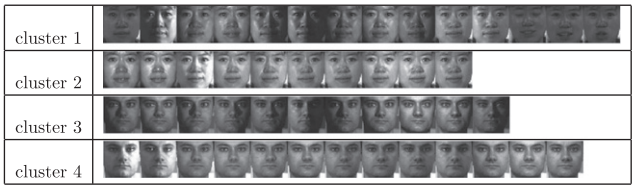
Fig. 2. Clustering results of SC^{3}SR, FCM and SCC. (a) SC^{3}SR1. (b) SC^{3}SR2. (c) FCM1. (d) FCM2. (e) SCC1. (f) SCC2.

And thanks to the suitable clustering results, both SC^{3}SR and SCC achieve the 100% classification accuracy in both kernelized or unkernelized versions. By contrast, the clustering results of FCMs, shown in Fig. 2(c) and (d), are not consistent with the class labels. FCM1 groups the upper two Gaussian components into one cluster, and FCM2 cuts the upper two Gaussian components apart in the middle respectively. Both of them fail to reflect the true underlying data structure.

4.2. Effectiveness of clustering learning

We show the clustering results on CMU PIE face database to demonstrate the clustering effectiveness of SC^{3}SR algorithm. The original PIE database has 68 persons with 41,368 face images. Each person is imaged under 13 different poses, 43 different illumination conditions, and with 4 different expressions. Here we only choose a subset (Pose C27[1]) with frontal pose and varying illumination of two persons. All images are down-sampled to 16×16 pixels. The clustering results are shown in Tables 3–5. In the result of SC^{3}SR, the faces grouped into the same cluster are under the similar illumination condition. Furthermore, each cluster only contains one person's faces thanks to the guidance of label. In the result of FCM, the faces with similar illumination are grouped together, but due to the absence of label guidance, FCM mixes up the faces of different persons and groups them together in Cluster 1 and Cluster 2. We circumscribe faces of different persons with different colors in Table 4. In the result of SCC, the faces belonging to the same persons are clustered into their respective clusters. In this situation, the

Table 3
Clustering result on CMU-PIE face dataset of SC^{3}SR.



[1] http://www.zjucadcg.cn/dengcai/Data/PIE/Pose27_64x64.mat

objective of SCC reaches its minimum value zero. SCC traps into a bad local minimum in this dataset which fails to reveal the cluster structure within classes. Table 6 illustrates the probabilistic relation of clusters and classes produced by SC^{3}SR,FCM and SCC respectively.

4.3. Effectiveness of classification learning

We examine the accuracy of classifications of SC^{3}SR algorithm on the benchmark datasets. We use the same dataset with Cai et al.'s paper and follow their experiment setting [5]. We copy SCC's experiment results from Cai's paper into Table 7. In Cai et al.'s paper, SCC algorithm is compared with classical algorithm like SVM and the related algorithms (including RFRC, VQ+LVQ3, RBFNN, RBFNN_PSO) mentioned in the Review section, and achieves better classification performance. We here omit the experiment results of these algorithms due to table space limitation. Please refer to Cai's paper if the readers are interested.

For each type of algorithms, results of unkernelized and kernelized versions are reported. The algorithm names appended by 1 or 2 denote the unkernelized or kernelized versions respectively. RBF kernel is used in the kernelized algorithms, and the kernel parameter γ is set to the mean of Euclidean distance between each data point pairs. The kernelized SC^{3}SR2 algorithm employs the approximated empirical kernel mapping which maps the data into an approximated low dimensional kernel space, here the dimension is set to $\sqrt{N}$. The parameters λ_1 and λ_2 are determined by searching in {1e−3 1e−2 1e−1 1e0 1e1 1e2 1e3}, and the cluster number is determined by searching in {C 2C 3C}. 5-fold cross validation is used to choose the reasonable parameters. To be fair, kernelized LR2 also uses the approximated empirical kernel mapping with the same parameter setting as SC^{3}SR2. In all the experiments, each dataset is randomly partitioned into two halves, one for training and the other for testing. For each dataset, the algorithms are run repeatedly and independently ten times, and the mean accuracy and standard deviation is reported in Table 7. For SCC, we copy the results reported in [5].

First, we compare the performance of SC^{3}SR and its base logistic regression classifier. According to the second and sixth columns of Table 7, the performance of SC^{3}SR1 is better than LR1. SC^{3}SR1 yields higher accuracies than LR1 on 11 out of 19 datasets, and comparable accuracies on 7 datasets, and only yields lower accuracies on one dataset. According to the third and seventh columns of Table 7, SC^{3}SR2 performs consistently better than LR2, and achieves higher accuracies on 14 out of 19 datasets and comparable accuracies on the rest datasets. The experiments demonstrate that by adopting data cluster structure representation, the classification ability of SC^{3}SR is highly enhanced compared with LR algorithm which not adopt data cluster structures.

Second, we compare the performance of SC^{3}SR2 and SC^{3}SR1. According to the sixth and seventh columns of Table 7, SC^{3}SR2 obtains also slightly better performance than SC^{3}SR1. Among all 19 datasets, the accuracies of SC^{3}SR2 are higher than those of SC^{3}SR1 on 7 of them, and are comparable on 9 of them, and are lower on the rest 3 datasets. We can see that kernel does bring some merits on some datasets. However sometimes the performance is not promoted obviously due to its heuristic setting rather than exhaustive searching in some range for kernel parameter and due to the approximate empirical kernel mapping which maybe lead to some information loss.

Third, we compare performances of SC^{3}SR and SCC. For comparison fairness, we compare SC^{3}SR1,2 with SCC1,2 together rather than SC^{3}SR1 with SCC1 and SC^{3}SR2 with SCC2 respectively because in fact SCC1 is in nature a non-linear classifier while SC^{3}SR1 is a linear one. Over all the benchmark datasets, SC^{3}SR achieves higher classification accuracies on 8 datasets, comparable accuracies on 7 datasets and lower accuracies on the rest 4 datasets. So overall SC^{3}SR produces better classification performance than SCC. Though SC^{3}SR does not significantly outperform SCC on some datasets, it still provides us a new faster and more flexible way to conduct simultaneous learning clustering and classification tasks.

4.4. Running time comparison between SC^{3}SR and SCC

To exam the efficiency of SC^{3}SR and SCC, we compare the running times of SC^{3}SR and SCC on some chosen datasets with the scales range from small (less than 100) to large (more than 4000). All the experiments are run in matlab environment on a PC with Intel Core2 Duo E7500 2.93 GHz CPU and 2G memory. For each dataset, the algorithms run 10 rounds and the average running times are reported in Table 8. It is obvious that SC^{3}SR is much faster than SCC by almost an order of magnitude which verifies the theoretical time complexity analysis in Section 3.4.2. SCC clearly cannot efficiently handle large datasets.

Table 4
Clustering result on CMU-PIE face dataset of FCM.

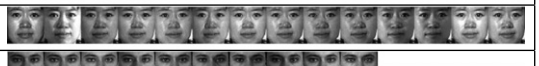
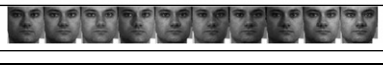
cluster 1	
cluster 2	
cluster 3	
cluster 4	

Table 6
Relation between clusters and classes of SC^{3}SR on CMU PIE face dataset.

Relation matrix of SC^{3}SR		*Cluster*1	*Cluster*2	*Cluster*3	*Cluster*4
	*Person*1	1	1	0	0
	*Person*2	0	0	1	1
Relation matrix of FCM		*Cluster*1	*Cluster*2	*Cluster*3	*Cluster*4
	*Person*1	0.5	0.3	1	0
	*Person*2	0.5	0.7	0	1
Relation matrix of SCC		*Cluster*1	*Cluster*2	*Cluster*3	*Cluster*4
	*Person*1	1	0	0	0
	*Person*2	0	0	0	1

Table 5
Clustering result on CMU-PIE face data set of SCC.

cluster 1	
cluster 2	N/A
cluster 3	N/A
cluster 4	

Table 7
Classification performance of logistic regression, SCC and SC^{3}SR. The best results are bolded. If there are more than one bold result in one dataset, it means that these results are comparative and have no significant difference.

Dataset	LR1	LR2	SCC1 [5]	SCC2 [5]	SC^{3}SR1	SC^{3}SR2
WBCD	95.7 ± 0.6	94.6 ± 1.4	$\mathbf{96.8 \pm 0.6}$	$\mathbf{97.0 \pm 0.4}$	$\mathbf{97.0 \pm 0.8}$	$\mathbf{96.8 \pm 1.2}$
Water	91.1 ± 3.7	95.9 ± 1.8	$\mathbf{98.3 \pm 1.5}$	$\mathbf{98.4 \pm 1.2}$	$\mathbf{98.0 \pm 1.0}$	$\mathbf{98.3 \pm 2.2}$
Thyroid	95.1 ± 1.6	91.3 ± 2.5	$\mathbf{96.3 \pm 1.5}$	$\mathbf{96.4 \pm 1.5}$	$\mathbf{95.5 \pm 1.7}$	95.2 ± 2.0
Lung cancer	43.5 ± 19.1	65.8 ± 9.9	48.3 ± 13.3	48.3 ± 14.2	48.8 ± 10.0	$\mathbf{75.8 \pm 5.8}$
PID	75.4 ± 1.3	75.3 ± 1.7	74.4 ± 2.3	$\mathbf{76.6 \pm 1.4}$	$\mathbf{77.4 \pm 1.1}$	75.4 ± 1.5
Soybean-small	97.1 ± 2.0	$\mathbf{99.1 \pm 1.7}$	96.5 ± 3.3	$\mathbf{99.6 \pm 1.3}$	97.5 ± 2.1	$\mathbf{100.0 \pm 0.0}$
WDBC	93.9 ± 1.5	95.0 ± 1.2	$\mathbf{96.6 \pm 0.9}$	$\mathbf{96.8 \pm 0.7}$	$\mathbf{97.1 \pm 0.8}$	$\mathbf{96.7 \pm 1.0}$
Ionosphere	85.3 ± 1.2	91.7 ± 2.6	92.1 ± 1.5	$\mathbf{93.2 \pm 1.4}$	85.7 ± 3.2	$\mathbf{93.2 \pm 1.3}$
Waveform	$\mathbf{86.8 \pm 1.0}$	85.4 ± 0.5	82.9 ± 2.4	$\mathbf{86.2 \pm 0.6}$	$\mathbf{86.5 \pm 0.6}$	$\mathbf{86.3 \pm 0.7}$
Balance scale	87.9 ± 1.1	92.4 ± 1.1	89.4 ± 1.6	90.6 ± 1.3	89.1 ± 0.6	$\mathbf{95.1 \pm 1.2}$
Heart disease	81.8 ± 1.9	79.6 ± 2.7	82.7 ± 1.9	$\mathbf{83.0 \pm 2.1}$	81.7 ± 2.5	81.2 ± 2.9
Glass	61.9 ± 4.6	59.8 ± 3.8	65.1 ± 3.6	64.9 ± 2.5	58.0 ± 4.6	$\mathbf{66.4 \pm 3.4}$
Sonar	69.2 ± 3.9	77.1 ± 3.0	$\mathbf{81.7 \pm 4.5}$	80.8 ± 5.1	74.5 ± 3.4	78.0 ± 5.0
Wine	95.3 ± 1.5	91.8 ± 3.1	96.9 ± 1.5	$\mathbf{97.1 \pm 1.8}$	$\mathbf{97.1 \pm 1.5}$	95.1 ± 2.3
Ecoli	71.3 ± 5.1	69.1 ± 4.9	82.9 ± 3.7	$\mathbf{83.7 \pm 1.8}$	71.2 ± 3.0	71.2 ± 2.1
Lenses	70.1 ± 18.4	71.5 ± 10.9	77.5 ± 9.9	77.5 ± 3.0	76.9 ± 7.2	$\mathbf{79.3 \pm 9.6}$
Iris	95.4 ± 1.0	93.1 ± 2.5	94.9 ± 1.0	95.2 ± 1.4	$\mathbf{96.5 \pm 1.2}$	95.2 ± 2.0
Bupa	66.9 ± 3.8	68.7 ± 2.4	64.2 ± 3.0	67.5 ± 5.8	68.5 ± 2.5	$\mathbf{69.2 \pm 2.3}$
Spambase	$\mathbf{91.8 \pm 0.5}$	$\mathbf{92.0 \pm 1.0}$	78.5 ± 7.7	88.1 ± 1.3	$\mathbf{91.9 \pm 1.5}$	$\mathbf{92.0 \pm 0.8}$

Table 8
Running time comparison between SC^{3}SR and SCC. All the experiments are run in matlab environment on a PC with Intel Core2 Duo E7500 2.93 GHz CPU and 2G memory.

Dataset (#instances × #dim × #class)	SC^{3}SR(s)	SCC(s)
Lenses ($24 \times 4 \times 3$)	0.72	16.64
Iris ($150 \times 4 \times 3$)	2.00	21.16
Sonar ($208 \times 60 \times 2$)	2.45	30.17
Bupa ($345 \times 6 \times 2$)	2.69	25.97
WDBC ($569 \times 30 \times 2$)	5.98	41.53
WBCD ($683 \times 9 \times 2$)	4.67	36.99
PID ($768 \times 8 \times 2$)	6.94	38.63
Spambase ($4601 \times 57 \times 2$)	52.93	454.75

4.5. Performance of SemiSC^{3}SR

In this subsection, we preliminarily examine the performance of SemiSC^{3}SR on 7 UCI data sets in this experiment to illustrate that our framework can be adapted to any semi-supervised scenario. The experiment setup follows Li et al. [16] and Mallapragada et al. [19]. Specifically, each data is split into two halves, with one half for training and the other for testing. For all the datasets, 10 random instances in the training set are labeled and the rest are used as unlabeled instances. The experiment is repeated 20 times and their average results are reported. Throughout the experiment, linear kernel is used. Parameters of SemiSC^{3}SR are set as $\lambda_1 = 1e3, \lambda_2 = 1e{-}1, \lambda_3 = 1e{-}1$, and parameters of SC^{3}SR are set as $\lambda_1 = 1e3, \lambda_2 = 1e{-}1$. We compare with both SVM and lapSVM, and directly copy their results in Li et al. and Mallapragada et al.'s papers into Table 9. In addition we also add another semi-supervised lapRLS [1] as a baseline.

According to Table 9, generally the semi-supervised algorithms outperform the supervised algorithms learnt just from a few given labeled instances. By comparison with SC^{3}SR, SemiSC^{3}SR obtains six higher accuracies on seven data sets. Unlabeled data as well as manifold regularization can indeed help classification for these data sets. On 7 data sets, SemiSC^{3}SR wins on 3, and both lapSVM and lapRLS wins on 2 respectively. Generally their performances are comparable, SemiSC^{3}SR's performance is slightly better.

Table 9
Accuracy comparison on UCI data sets. The highest accuracy is bolded.

Dataset	SVM [16,19]	lapSVM [16,19]	lapRLS	SemiSC^{3}SR	SC^{3}SR
House	91.16	89.95	87.90	**92.33**	89.51
Heart	70.59	77.96	**78.11**	71.82	68.76
Vehicle	78.28	71.38	72.53	**78.40**	76.50
dbc	75.74	**91.07**	89.59	86.86	84.92
isolet	89.58	**93.93**	93.92	87.78	93.97
optdigits	90.31	98.34	**98.75**	96.90	96.32
ethn	67.04	74.60	73.51	**75.36**	67.53

5. Conclusion

In this paper, a new simultaneous learning clustering and classification framework SC^{3}SR is presented. In this framework, cluster structure representation is proposed to bridge the clustering and classification tasks, so the performances of clustering and classification tasks could benefit from each other. This is different with the SCC framework proposed by Cai et al. [5] which models cluster posterior probabilities of classes and bridges the clustering and classification tasks by Bayesian formula. Comparing with SCC, the formulation of SC^{3}SR is easier to optimize. The block coordinate descent method is used to optimize SC^{3}SR under guaranteed convergence. It is worth mentioning that extending SCC to semi-supervised scenario is not straightforward [5] because for unlabeled data, $p(class|cluster)$ cannot be estimated, however extending SC^{3}SR to semi-supervised scenario is relatively easier by the idea from manifold regularization framework [1]. Furthermore, SC^{3}SR can also perform dimension reduction tasks. The learned low dimensional cluster structure representation can naturally reflect the original data structure. However, it is not so obvious for SCC to generate dimension reduced data representation. In the next step, we want to examine the influence of different CSRs on SC^{3}SR framework.

Acknowledgment

We thank National Science Foundation of China under Grant nos. 61035003, 60973097 and 61003116 respectively and National Science Foundation of Jiangsu Province under Grant no. BK2010263 for partial support.

Appendix A

In this appendix, we give out the gradients and Hessian matrices used in SC^{3}SR and SemiSC^{3}SR algorithm. In the following, the notation $\odot$ and $\oslash$ means elementwise product and divide. For notation convenience, we let $p(c=m|x/h_i)=p_{mi}^{x/h}$ and $p_{m:}^{x/h}=[p_{m1}^{x/h},\ldots,p_{mN}^{x/h}]$.

A.1. Gradients and Hessian matrices used in SC^{3}SR

The gradient of W^x is $g_{W_x}=[g_{w_1^x}^T\cdots g_{w_{C-1}^x}^T]^T$ with $g_{w_m^x}$ defined as follows:

$$g_{w_m^x}=\lambda_1\left(\sum_i p_{mi}^x x_i-\sum_{y_i==m}x_i\right)+\lambda_2\sum_i\left(p_{mi}^x(1-p_{mi}^x)\left(1+\log\left(\frac{p_{mi}^x}{p_{mi}^h}\right)\right)-p_{mi}^h(1-p_{mi}^x)\right)x_i+\lambda_2\sum_i\sum_{c\neq m}\left(-p_{ci}^h p_{mi}^x\left(1+\log\left(\frac{p_{mi}^x}{p_{mi}^h}\right)\right)+p_{ci}^h p_{mi}^h\right)x_i$$

and the Hessian of W_x is

$$H_{W_x}=\begin{pmatrix} XD_{1,1}^xX^T & \cdots & XD_{1,C-1}^xX^T\\ & \ddots & \\ XD_{C-1,1}^xX^T & \cdots & XD_{C-1,C-1}^xX^T\end{pmatrix}$$

where $D_{m,n}^x$ is a diagonal matrix with diagonal elements in $d_{m,n}^x$ defined as follows:

$$\begin{aligned}d_{m,m}^x&=\lambda_1 p_{m:}^x\odot(1-p_{m:}^x)\\&+\frac{1}{2}\lambda_2(p_{m:}^x\odot(1-p_{m:}^x)\odot(1-2p_{m:}^x)\odot(1+\log(p_{m:}^x\oslash p_{m:}^h))\\&+p_{m:}^x\odot(1-p_{m:}^x)(1-p_{m:}^x+p_{m:}))\\&+\frac{1}{2}\lambda_2\sum_{c\neq m}(-p_{c:}\odot p_{m:}^x\odot(1-2p_{m:}^x)\odot(1+\log(p_{c:}^x\oslash p_{c:}^h))\\&+p_{c:}^x\odot {p_{m:}^x}^{.2}+p_{c:}^h\odot p_{m:}^x\odot(1-p_{m:}^x))\end{aligned}$$

$$\begin{aligned}d_{m,n}^x&=\lambda_1 p_{m:}^x\odot p_{n:}^x+\frac{1}{2}\lambda_2(p_{m:}^x\odot p_{n:}^x\odot(2p_{m:}^x-1)\odot(1+\log(p_{m:}^x\oslash p_{m:}^h))\\&-p_{m:}^x\odot p_{n:}^x(1-p_{m:}^x+p_{m:}^h))+\frac{1}{2}\lambda_2(p_{n:}^x\odot p_{m:}^x\odot(2p_{n:}^x-1)\\&\odot(1+\log(p_{n:}^x\oslash p_{n:}^h))-p_{n:}^x\odot p_{m:}^x(1-p_{n:}^x+p_{n:}^h))\\&+\frac{1}{2}\lambda_2\sum_{c\neq m,c\neq n}(p_{c:}^x\odot p_{m:}^x\odot p_{n:}^x(1.5\log(p_{c:}^x\oslash p_{c:}^h))\\&-p_{c:}^h\odot p_{m:}^x\odot p_{n:})\end{aligned}$$

Consider the symmetry of W_x and W_h in the objective, the gradient and Hessian of W_h can be derived by simply exchanging x with h in the W_x's equations. We omit the lengthy mathematical deduction here.

The gradient of h_i for $i=1,\ldots,N$ is defined as follows:

$$g_{h_i}=\frac{1}{2}\sum_k\|x_i-v_k\|^2h_i+\lambda_1(w_{y_i}^h-\psi_i)+\frac{1}{2}\lambda_2\sum_c((\log(p_{ci}^h)+1-p_{ci}^x/p_{ci}^h-\log(p_{ci}^x))p_{ci}^hw_c^h-\psi_i)$$

where $\psi_i=\sum_c p_{ci}^hw_c^h/\sum_c p_{ci}^h$.

A.2. Gradient and Hessian matrices used in SemiSC^{3}SR

Comparing with SC^{3}SR's objective, the objective of SemiSC^{3}SR has one more λ_3 term. Thus by adding the gradients and Hessian matrices of the λ_3 term to those of SC^{3}SR, we get the corresponding SemiSC^{3}SR's. We give out the gradients and Hessian matrices, marked by tilde, of the λ_3 term here.

For notation convenience, we let $\log(p(c=m|(x/h)_i))=q_{mi}^{x/h}$ and $\log(p_{m:}^{x/h})=[q_{m1}^{x/h},\ldots,q_{mN}^{x/h}]$. W is the graph weight matrix and $L=T-W$ is the Laplacian matrix where T is diagonal matrix with $T_{ii}=\sum_j W_{ij}$. Let $t=[T_{11},\cdot,T_{NN}]$. L_i is the ith column of Laplacian matrix L and $e=[1,\ldots,1]$.

The gradient of λ_3 term with respect to W_x is $\tilde{g}_{W_x}=[\tilde{g}_{w_1^x}^T\cdots\tilde{g}_{w_{C-1}^x}^T]^T$ with $\tilde{g}_{w_m^x}$ defined as follows:

$$\tilde{g}_{w_m^x}=4\lambda_3\left(\sum_i(1-p_{mi}^x)(q_{m:}^xL_i)x_i\right)+4\lambda_3\left(-\sum_i(p_{mi}^x(q_{m:}^xL_i)x_i\right)$$

and the Hessian of λ_3 term with respect to W_x is

$$\tilde{H}_{W_x}=\lambda_3\begin{pmatrix} X(\tilde{D}_{1,1}^x+\tilde{B}_{1,1}^x)X^T & \cdots & X(\tilde{D}_{1,C-1}^x+\tilde{B}_{1,C-1}^x)X^T\\ & \ddots & \\ X(\tilde{D}_{C-1,1}^x+\tilde{B}_{C-1,1}^x)X^T & \cdots & X(\tilde{D}_{C-1,C-1}+\tilde{B}_{C-1,C-1}^x)X^T\end{pmatrix}$$

where $\tilde{D}_{m,n}^x$ is a diagonal matrix with diagonal elements in $\tilde{d}_{m,n}^x$, and $\tilde{d}_{m,n}^x$ and $\tilde{B}_{m,n}^x$ is defined as follows:

$$\begin{aligned}\tilde{d}_{m,m}^x&=-4\sum_c p_{m:}^x\odot(1-p_{m:}^x)\odot(q_{c:}^xL)\\&+4\sum_{c\neq m}t\odot p_{m:}^x\odot p_{m:}^x-4t\odot(1-p_{m:}^x)\odot(1-p_{m:}^x)\end{aligned}$$

$$\tilde{B}_{m,m}^x=\sum_{c\neq m}4W\odot(p_{m:}^xTp_{m:}^x)-4W\odot((1-p_{m:}^x)^T(1-p_{m:}))$$

$$\begin{aligned}\tilde{d}_{m,n}&=4\sum_c p_{m:}^x\odot p_{n:}^x\odot(q_{c:}L)\\&+4\sum_{c\neq m,n}t\odot p_{m:}^x\odot p_{n:}^x+4t\odot((1-p_{m:}^x)\odot p_{n:}^x+(1-p_{n:}^x)\odot p_{m:}^x)\end{aligned}$$

$$\tilde{B}_{m,n}=-4CW\odot(p_{m:}^xTp_{n:}^x)+4(e^Tp_{m:}^x+p_{n:}^xTe)$$

Consider the symmetry of W_x and W_h in λ_3 term, the gradient and Hessian of W_h could be derived by simply exchange x with h in the W_x's equations. We also omit the lengthy mathematical equations here.

The gradient of λ_3 term with respect to h_i for $i=1,\ldots,N$ defined as follows:

$$\tilde{g}_{h_i}=\sum_c^{C-1}q_{c:}^hL_i(w_c^h-\tau_i)-q_{C:}^hL_i\tau_i$$

where $\tau_i=\sum_c^{C-1}p_{ci}^hw_c^h$.

References

[1] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, The Journal of Machine Learning Research 7 (2006) 2399–2434.

[2] D.P. Bertsekas, W.W. Hager, O.L. Mangasarian, Nonlinear Programming, Athena Scientific Belmont, MA, 1999.

[3] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2006.

[4] W. Cai, S. Chen, D. Zhang, Robust fuzzy relational classifier incorporating the soft class labels, Pattern Recognition Letters 28 (16) (2007) 2250–2263.

[5] W. Cai, S. Chen, D. Zhang, A simultaneous learning framework for clustering and classification, Pattern Recognition 42 (7) (2009) 1248–1259.

[6] W. Cai, S. Chen, D. Zhang, A multiobjective simultaneous learning framework for clustering and classification, IEEE Transactions on Neural Networks 21 (2) (2010) 185–200.

[7] O. Chapelle, B. Scholkopf, A. Zien, et al., Semi-supervised Learning, MIT Press, 2006.

[8] S.C. Chen, D.Q. Zhang, A novel kernelized fuzzy c-means algorithm with application in medical image segmentation, Artificial Intelligence in Medicine 32 (2004) 37–50.

[9] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley, New York, 2001.

[10] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice-Hall, 1994.
[11] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (1) (2000) 4–37.
[12] M.R. Gray, J.M. Keller, J.R. James, A. Givens, A fuzzy k-nearest neighbor algorithm, IEEE Transactions on Systems, Man, and Cybernetics 15 (4) (1985) 581.
[13] S.W. Kim, B.J. Oommen, Enhancing prototype reduction schemes with LVQ3-type algorithms, Pattern Recognition 36 (5) (2003) 1083–1093.
[14] X. Li, N. Ye, Grid-and dummy-cluster-based learning of normal and intrusive clusters for computer intrusion detection, Quality and Reliability Engineering International 18 (3) (2002) 231–242.
[15] X. Li, N. Ye, A supervised clustering and classification algorithm for mining data with mixed variables, IEEE Transactions on Systems, Man and Cybernetics, Part A 36 (2) (2006) 396–406.
[16] Y.F. Li, J.T. Kwok, Z.H. Zhou, Semi-supervised learning using label mean, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 633–640.
[17] J. Liu, J. Ye, Efficient Euclidean projections in linear time, in: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 657–664.
[18] I. Maglogiannis, H. Sarimveis, C.T. Kiranoudis, A.A. Chatziioannou, N. Oikonomou, V. Aidinis, Radial basis function neural networks classification for the recognition of idiopathic pulmonary fibrosis in microscopic images, IEEE Transactions on Information Technology in Biomedicine 12 (1) (2008) 42–54.
[19] P.K. Mallapragada, R. Jin, A.K. Jain, Y. Liu, Semiboost: boosting for semi-supervised learning, IEEE Transactions on Pattern Analysis and Machine Intelligence (2008) 2000–2014.
[20] L. Qiao, L. Zhang, S. Chen, An empirical study of two typical locality preserving linear discriminant analysis methods, NeuroComputing 73 (10–12) (2010) 1587–1594.
[21] B. Scholkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.R. Muller, G. Ratsch, A.J. Smola, Input space versus feature space in kernel-based methods, IEEE Transactions on Neural Networks 10 (5) (2002) 1000–1017.
[22] M. Setnes, R. Babuska, Fuzzy relational classifier trained by fuzzy clustering, IEEE Transactions on Systems, Man, and Cybernetics, Part B 29 (5) (2002) 619–625.
[23] H. Xiong, M.N.S. Swamy, M.O. Ahmad, Optimizing the kernel in the empirical feature space, IEEE Transactions on Neural Networks 16 (2) (2005) 460–474.
[24] H. Xue, S. Chen, Q. Yang, Discriminatively regularized least-squares classification, Pattern Recognition 42 (1) (2009) 93–104.
[25] Z.R. Yang, A novel radial basis function neural network for discriminant analysis, IEEE Transactions on Neural Networks 17 (3) (2006) 604–612.
[26] N. Ye, X. Li, A scalable, incremental learning algorithm for classification problems, Computers and Industrial Engineering 43 (4) (2002) 677–692.

Qiang Qian received a BSc from Nanjing University of Aeronautics and Astronautics (NUAA), PR China. Currently he is a PhD Student at the Department of Computer Science and Engineering, NUAA. His research interests include data mining and pattern recognition.

Songcan Chen received the BSc degree in Mathematics from Hangzhou University (now merged into Zhejiang University) in 1983. In December 1985, he completed the MSc degree in Computer Applications at Shanghai Jiaotong University and then worked at Nanjing University of Aeronautics and Astronautics (NUAA) in January 1986 as an Assistant Lecturer. There he received a PhD degree in Communication and Information Systems in 1997. Since 1998, as a full Professor, he has been with the Department of Computer Science and Engineering at NUAA. His research interests include pattern recognition, machine learning and neural computing. In these fields, he has authored or coauthored over 130 scientific journal papers.

Weiling Cai received a BSc and PhD degrees in Computer Science from Nanjing University of Aeronautics and Astronautics, China, in 2003 and 2008, respectively. At present, she is a lecturer at the Department of Computer Science & Technology, Nanjing Normal University. Her research interests focus on Machine Learning and Pattern Recognition.