Pattern Recognition Letters 98 (2017) 90–95

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Semi-supervised manifold regularization with adaptive graph construction

Yunyun Wang[a,b,*], Yan Meng[a], Yun Li[a,b], Songcan Chen[c], Zhenyong Fu[a], Hui Xue[d]

[a] Department of Computer Science and Engineering, Nanjing University of Posts & Telecommunications, Nanjing 210046, P.R. China
[b] Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, Nanjing, Jiangsu 210023, China
[c] Department of Computer Science and Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, P.R. China
[d] School of Computer Science and Engineering, Southeast University, Nanjing 210096, P.R. China

ARTICLE INFO

Article history:
Received 8 September 2016
Available online 5 September 2017

Keywords:
Semi-supervised classification
Manifold regularization
Manifold graph
Graph construction

ABSTRACT

Manifold regularization (MR) provides a powerful framework for semi-supervised classification (SSC) learning. It imposes the smoothness constraint over a constructed manifold graph, and its performance largely depends on such graph. However, 1) The manifold graph is usually pre-constructed before classification, and fixed during the classification learning process. As a result, independent with the subsequent classification, the graph does not necessarily benefit the classification performance. 2) There are parameters needing tuning in the graph construction, while parameter selection in semi-supervised learning is still an open problem currently, which sets up another barrier for constructing a "well-performing" manifold graph benefiting the performance. To address those issues, we develop a novel semi-supervised manifold regularization with adaptive graph (AGMR for short) in this paper by integrating the graph construction and classification learning into a unified framework. In this way, the manifold graph along with its parameters will be optimized in learning rather than pre-defined, consequently, it will be adaptive to the classification, and benefit the performance. Further, by adopting the entropy and sparse constraints respectively for the graph weights, we derive two specific methods called AGMR_entropy and AGMR_sparse, respectively. Our empirical results show the competitiveness of those AGMRs compared to MR and some of its variants.

1. Introduction

In many real applications, unlabelled data can be easily and cheaply collected, while the acquisition of labelled data is usually quite expensive and time-consuming, especially involving manual effort, e.g., in web page recommendation and spam email detection. Consequently, semi-supervised classification, which exploits a large amount of unlabelled data jointly with the limited labelled data for classification learning, has attracted intensive attention during the past decades [7,25,26,28].

Generally, semi-supervised classification methods attempt to exploit the intrinsic data distribution information disclosed by the unlabeled data in learning. To exploit the unlabeled data, some assumption should be adopted for learning. Two common assumptions in semi-supervised classification are the cluster assumption and the manifold assumption [7,19,26]. The former assumes that similar instances are likely to share the same class label, thus guides the classification boundary passing through the low density region between clusters. The latter assumes that the data are resided on some low dimensional manifold represented by a Laplacian graph, and similar instances should share similar classification outputs according to the graph. Almost all off-the-shelf semi-supervised classification methods adopt one or both of those assumptions explicitly or implicitly [7,25]. For instance, the large margin semi-supervised classification methods, such as Transductive Support Vector Machine (TSVM) [15], semi-supervised SVM (S3VM) [11] and their variants [8,17], adopt the cluster assumption. The graph-based semi-supervised classification methods, such as label propagation [4,27], graph cuts [5] and manifold regularization (MR) [3], adopt the manifold assumption.

The graph-based semi-supervised classification methods are mainly transductive ones, except MR. Although transductive methods have specific applications, many real tasks need predicting unseen instances, thus need inductive methods. As a result, as an inductive graph-based semi-supervised classification method, MR has attracted much attention and applied in many learning tasks such as image retrieval [14] and web spam identification [1], etc. In this paper, we will concentrate on the MR framework [3].

* Corresponding author.
E-mail address: wangyunyun@njupt.edu.cn (Y. Wang).

http://dx.doi.org/10.1016/j.patrec.2017.09.004
0167-8655/

The learning process of MR includes two steps: First, a manifold graph is constructed to describe the manifold structure of instances, in which the graph nodes represent instances, and the weights represent the similarities between instances. Then, according to the manifold assumption, the smoothness constraint over the constructed graph is implemented in terms of regularization. The construction of manifold graph is critical for the performance of MR. Once a "well-performing" graph benefiting the subsequent classification is constructed, it can finally help boost the classification performance. Otherwise, it will not help the classification, or even hurt the performance. However, on one hand, the graph is usually defined in advance and kept fixed during the learning process. It is actually impossible for us to judge whether a graph is a "well-performing" graph in advance. As a result, it is really difficult to construct a "well-performing" graph before classification. On the other hand, there are parameters needing tuning in the manifold graph, whereas in semi-supervised learning with limited label information, the parameter selection is still an open problem with no effective solution yet. It sets up another barrier for graph constructing for MR in advance. As far as we know, the existing improvements of MR either attempt to select the regularization parameters [12], or try to improve the efficiency of MR [23,21], few researches have concentrated on graph construction up to now. Actually, the graph learning issue is considered as a separate topic under research currently, although the adaptive graph construction has been studied in GoLPP [24] for dimension reduction, MR and its improvements mainly adopt a pre-constructed graph.

To address the above two issues, we aim to develop a new MR framework for semi-supervised classification here by introducing an adaptive graph (AGMR for short). In AGMR, the construction of manifold graph is incorporated into the classification learning. In this way, the manifold graph along with its parameters can be automatically adjusted in learning rather than specified in advance. The graph construction and classification learning are combined together, thus can be more likely to benefit each other. Further, by adopting the entropy and sparse constraints for the graph weights, respectively, we derive two specific methods called AGMR_entropy and AGMR_sparse, respectively. The implementation follows an alternating iterative strategy to optimize the decision function and the manifold graph, respectively. Each step in the iteration results in a closed-form solution, and its iterative convergence can theoretically be guaranteed. Experiments on several real datasets show the competitive performance of AGMR compared with MR and its improvements with different graph constructed.

The rest of this paper is organized as follows. Section 2 introduces the related works, Section 3 presents the proposed graph-adaptive MR framework, Section 4 gives the empirical results, and some conclusions are drawn in Section 5.

2. Related works

Given labeled data $X_l = \{x_i\}_{i=1}^{l}$ with the corresponding labels $Y = \{y_i\}_{i=1}^{l}$, and unlabeled data $X_u = \{x_j\}_{j=l+1}^{n}$, where each $x_i \in R^d$ and $u = n\text{-}l$. $G = \{w_{ij}\}_{i,j=1}^{n}$ is a pre-specified Laplacian graph over the whole dataset, where each weight w_{ij} represents the similarity between the connected instances x_i and x_j. There are two ways for deciding whether x_i and x_j are connected. One is the k-nearest neighbor strategy, i.e., x_i and x_j are connected if x_i is in the k-nearest neighbor of x_j (or x_j is in the k-nearest neighbor of x_i). The other is the ε-ball nearest neighbor strategy, i.e., x_i and x_j are connected when $\|x_i - x_j\|^2 < \varepsilon$. The weights over the graph describe the similarities between the connected instances, and can be specified by several weighting strategies. For example, the 0–1 weighting, i.e., $w_{ij} = 1$ if x_i and x_j are connected by an edge over the graph, the heat kernel weighing with $w_{ij} = e^{-\frac{\|x_i - x_j\|^2}{\sigma}}$ if x_i and x_j are connected, or the dot-product weighting with $w_{ij} = x_i^T x_j$ if x_i and x_j are connected.

After the construction of the manifold graph, the framework of MR can be formulated as follows with a decision function $f(x)$,

$$\min_f \sum_{i,j=1}^{l+u} w_{ij}(f(x_i) - f(x_j))^2 + \gamma_1 \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_2 \|f\|_K^2 \tag{1}$$

where $V(x_i, y_i, f)$ is the loss function, such as the hinge loss max $\{0, 1\text{-}y_i f(x_i)\}$ for support vector machine (SVM), or the square loss $(y_i - f(x_i))^2$ for regularized least square classifier (RLSC), in this way, the MR framework naturally embodies the specific algorithms LapSVM and LapRLSC [3]. $\|f\|_K^2$ is a regularization term for smoothness in the Reproducing Kernel Hilbert Space (RKHS). The third term guarantees the prediction smoothness over the manifold graph, which can be further written as

$$\sum_{i,j=1}^{l+u} w_{ij}(f(x_i) - f(x_j))^2 = 2\mathbf{f}^T\mathbf{L}\mathbf{f} \tag{2}$$

where $\mathbf{f} = [f(x_1), \ldots, f(x_{l+u})]^T$, and $\mathbf{L}$ is the graph Laplacian given by $\mathbf{L} = \mathbf{D}\text{-}\mathbf{W}$, $\mathbf{W}$ is the weight matrix of graph G and $\mathbf{D}$ is a diagonal matrix with the diagonal component given by $\mathbf{D}_{ii} = \sum_{j=1}^{n} w_{ij}$. According to the Representer theorem [3], the minimizer of problem (1) has the form

$$f^*(x) = \sum_{i=1}^{l+u} \alpha_i K(x_i, x) \tag{3}$$

where $K: X \times X \to R$ is a Mercer kernel (the bias of the decision function can be omitted by augmenting each instance with a 1-valued element).

3. Semi-supervised manifold regularization with adaptive graph (AGMR)

3.1. Model formulation

Given labeled instances $X_l = \{x_i\}_{i=1}^{l}$ with the corresponding labels $Y = \{y_i\}_{i=1}^{l}$, and unlabelled instances $X_u = \{x_j\}_{j=l+1}^{n}$, where each $x_i \in R^d$ and $u = n\text{-}l$. The optimization problem of AGMR can be formulated as

$$\begin{aligned} &\min_{f, w_{ij}} \sum_{i,j=1}^{l+u} w_{ij}(f(x_i) - f(x_j))^2 \\ &+\gamma_1 \sum_{i=1}^{l} (f(x_i) - y_i)^2 + \gamma_2 \|f\|_K^2 + \eta R(w_{ij}) \\ &s.t. \sum_{j=1}^{u} w_{ij} = 1 \\ &w_{ij} \geq 0 \end{aligned} \tag{4}$$

The first three terms in the optimization function of (4) are the same with those in MR, $R(w_{ij})$ is some constraint on the graph weights, and η is the regularization parameter. Different from MR seeking the decision function in learning, AGMR seeks both the decision function and the weights for the manifold graph. From the optimization problem in (4), we can find that: 1) The manifold graph in MR is specified before classification, and fixed in the learning process. While in AGMR, the graph is actually optimized in the learning process along with its parameters; 2) In AGMR, we have $\sum_{j=1}^{u} w_{ij} = 1$ and $w_{ij} \geq 0$, in this way, each w_{ij} actually reflects the probability that x_i and x_j should be in the same class; 3) Without the constraint $R(w_{ij})$ on each w_{ij}, the solution for each w_{ij} will degenerate to a trivial one, in which only one element is 1, and the remainder are all 0.

Different constraints for the graph weights generate different models, thus yield different classification performance. In the following, we will respectively use the entropy constraint and the sparse constraint for examples to develop new AGMR methods within the above framework.

3.2. AGMR with entropy constraint (AGMR_entropy)

3.2.1. Model formulation

To control the uniformity level of the manifold graph weights, we use an entropy regularization term for $R(w_{ij})$. $R(w_{ij}) = -\sum_{i=1}^{n} H(w_{i\cdot})$ where $w_{i\cdot} = [w_{i1}, w_{i2}, \ldots w_{in}] \in R^n$, and $H(w_{i\cdot})$ is the generalized entropy describing the uniformity level of each w_{ij}, $j = 1 \ldots n$. The generalized entropy has several versions, such as the celebrated Shannon entropy $H(v) = \sum_{i=1}^{n} -v_i \log v_i$, the L_1-entropy $H(v) = 2 - \sum_{i=1}^{n} |v_i - \frac{1}{n}|$, the L_2-entropy $H(v) = 1 - v^T v$ and the L_∞-entropy $H(v) = 1 - \max_{1 \le i \le n} v_i$. Different entropy versions yield different models, and we will adopt the celebrated Shannon entropy here for example.

Adopting celebrated Shannon entropy as an example, i.e., $H(w_{i\cdot}) = \sum_{j=1}^{n} -w_{ij} \log w_{ij}$, and the square loss function, the optimization can be further written as

$$\begin{aligned} &\min_{f, w_{ij}} \sum_{i,j=1}^{l+u} w_{ij}(f(x_i) - f(x_j))^2 + \gamma_1 \sum_{i=1}^{l} (f(x_i) - y_i)^2 \\ &+ \gamma_2 \|f\|_K^2 + \eta \sum_{i,j=1}^{n} w_{ij} \ln w_{ij} \\ &s.t. \sum_{j=1}^{u} w_{ij} = 1 \\ &w_{ij} \ge 0 \end{aligned} \tag{5}$$

By introducing the entropy constraint, the optimization problem in (5) actually adopts an Entropy Maximization criterion [6,10], which imposes a uniform distribution for the weights of the manifold graph to avoid a trivial solution.

3.2.2. Problem solution

The optimization problem of AGMR_entropy is non-convex with respect to (f, w_{ij}), and we will resort to the alternating iterative strategy to seek the decision function $f(x)$ and graph weights w_{ij} respectively. Fortunately, each step has a closed-form solution.

With fixed w_{ij}, the optimization problem of AGMR_entropy is actually the same with MR, which can be written as

$$\min_{f} \sum_{i,j=1}^{l+u} w_{ij}(f(x_i) - f(x_j))^2 + \gamma_1 \sum_{i=1}^{l} (f(x_i) - y_i)^2 + \gamma_2 \|f\|_K^2 \tag{6}$$

As in MR, the minimizer can also be formulated as $f^*(x) = \sum_{i=1}^{l+u} \alpha_i K(x_i, x)$ according to the Representation Theorem, and the solution is

$$\alpha = \left(\gamma_1 \mathbf{K}_l^T \mathbf{K}_l + \gamma_2 \mathbf{K} + \mathbf{K}^T \mathbf{L} \mathbf{K}\right)^{-1} \gamma_1 \mathbf{K}_l^T \mathbf{Y} \tag{7}$$

where $\alpha = [\alpha_1, \ldots \alpha_{l+u}]^T$ is the vector of Lagrange multipliers. $\mathbf{K}_l = (\mathbf{X}_l, \mathbf{X})_H \in R^{l \times (l+u)}$ and $\mathbf{K} = (\mathbf{X}, \mathbf{X})_H \in R^{(l+u) \times (l+u)}$ are the kernel matrices, where $\mathbf{X}_l$ and $\mathbf{X}$ denote the labeled and the whole datasets, respectively. $\mathbf{Y} = [y_1, \ldots y_l]^T$ is the vector of class labels for the labeled data.

With fixed $f(x)$, the optimization problem for w_{ij} can be written as

$$\begin{aligned} &\min_{w_{ij}} \sum_{i,j=1}^{n} w_{ij}\left(f(x_i) - f(x_j)\right)^2 + \eta \sum_{i,j=1}^{n} w_{ij} \ln w_{ij} \\ &s.t. \sum_{j=1}^{u} w_{ij} = 1 \\ &w_{ij} \ge 0 \end{aligned} \tag{8}$$

Using the Lagrange multiplier method, the solution for each w_{ij} can be written as (the details are given in Appendix A)

$$w_{ij} = \frac{e^{-\left(f(x_i) - f(x_j)\right)^2 / \eta}}{\sum_{j=1}^{n} e^{-\left(f(x_i) - f(x_j)\right)^2 / \eta}} \tag{9}$$

3.3. AGMR with sparse constraint (AGMR_sparse)

3.3.1. Model formulation

In fact, each instance in the manifold graph is connected with only a few neighbor instances, thus only a few elements of each weight $w_{i\cdot}$ should be non-zeros, and the rests should be zeros. That is, each weight vector $w_{i\cdot}$ should be sparse. As a result, we incorporate the sparse constraint $R(w_{ij}) = \eta_1 \sum_{i=1}^{u} \left(x_i - \sum_{j=1}^{u} w_{ij} x_j\right)^2 + \eta_2 \sum_{i=1}^{l+u} \|w_{i\cdot}\|_1$ into MR, where $\|\cdot\|_1$ is the l_1-norm, η_1 and η_2 are regularization parameters. Finally, the optimization problem can be formulated as

$$\begin{aligned} &\min_{f, w_{ij}} \sum_{i,j=1}^{l+u} w_{ij}(f(x_i) - f(x_j))^2 + \gamma_1 \sum_{i=1}^{l} (f(x_i) - y_i)^2 \\ &+ \gamma_2 \|f\|_K^2 + \eta_1 \sum_{i=1}^{u} \left(x_i - \sum_{j=1}^{u} w_{ij} x_j\right)^2 + \eta_2 \sum_{i=1}^{l+u} \|w_{i\cdot}\|_1 \\ &s.t. \sum_{j=1}^{u} w_{ij} = 1 \\ &w_{ij} \ge 0 \end{aligned} \tag{10}$$

3.3.2. Optimization solution

The optimization problem of AGMR_sparse is also non-convex with respect to (f, w_{ij}). We will resort to the alternating iterative strategy to seek $f(x)$ and w_{ij} respectively, each step has a closed-form solution.

With fixed w_{ij}, the optimization problem of AGMR_sparse is also the same with MR, i.e.,

$$\min_{f} \sum_{i,j=1}^{l+u} w_{ij}(f(x_i) - f(x_j))^2 + \gamma_1 \sum_{i=1}^{l} (f(x_i) - y_i)^2 + \gamma_2 \|f\|_K^2 \tag{11}$$

and the solution is already given in (7).

With fixed $f(x)$, the optimization problem for w_{ij} can be written as

$$\begin{aligned} &\min_{w_{ij}} \sum_{i,j=1}^{n} w_{ij}\left(f(x_i) - f(x_j)\right)^2 + \eta_1 \sum_{i=1}^{u} \left(x_i - \sum_{j=1}^{u} w_{ij} x_j\right)^2 \\ &+ \eta_2 \sum_{i=1}^{l+u} \|w_{i\cdot}\|_1 \\ &s.t. \sum_{j=1}^{u} w_{ij} = 1 \\ &w_{ij} \ge 0 \end{aligned} \tag{12}$$

the solution of w_{ij}, we adopt a strategy of auxiliary function [9,20], finally the solution can be formulated as (the details are given in Appendix B)

$$w_{ki} = w_{ki} \frac{\left(\eta_1 \mathbf{X}^T \mathbf{X} + \frac{1}{2} \mathbf{F}\mathbf{F}^T\right)_{ki}}{(\eta_1 \mathbf{X}^T \mathbf{X} \mathbf{W})_{ki} + \frac{1}{4}\left(\left(\mathbf{F}\mathbf{F}^T\right)_{ii} + \left(\mathbf{F}\mathbf{F}^T\right)_{kk}\right) + \frac{\eta_2}{2}} \tag{13}$$

where $\mathbf{X}$ is the data matrix, i.e., $\mathbf{X} = [x_1, x_2, \ldots x_n] \in R^{d \times n}$, where $x_i \in R^d$ and n is the total number of instances. $\mathbf{F}$ is the column vector of classification scores for instances, i.e., $\mathbf{F} = [f(x_1), \ldots, f(x_n)]' \in R^n$, where $f(x_i)$ is the classification score for each instance x_i.

3.4. Algorithm description

The optimization of AGMR (including both AGMR_entropy and AGMR_sparse) follows an alternating iterative strategy and the iteration starts from an initial $f(x)$ by MR. The iteration terminates when $|J^k - J^{k-1}| < \varepsilon J^{k-1}$, where J^k is the objective function value at the kth iteration and ε is a pre-defined threshold. The AGMR algorithm is described as follows,

Proposition 1. *The sequence $\{J(\boldsymbol{\alpha}^t, w_{ij}^t)\}$ obtained in the above algorithm w.r.t. AGMR converges.*

Proof. First, the sequence of the objective function values generated by the above algorithm decreases monotonically. In fact, the objective function $J(\boldsymbol{\alpha}, w_{ij})$ is biconvex [13] in $(\boldsymbol{\alpha}, w_{ij})$. Specifically, for fixed w_{ij}^t, the objective function is convex in $\boldsymbol{\alpha}$, thus the optimal $\boldsymbol{\alpha}^*$ can be obtained by minimizing $J(\boldsymbol{\alpha}, w_{ij}^t)$, or equivalently optimizing (6). Now set $\boldsymbol{\alpha}^{t+1} = \boldsymbol{\alpha}^*$, then $J(\boldsymbol{\alpha}^{t+1}, w_{ij}^t) = J(\boldsymbol{\alpha}^*, w_{ij}^t) \le J(\boldsymbol{\alpha}^t, w_{ij}^t)$. Simultaneously, with current $\boldsymbol{\alpha}^{t+1}$, the objective function is convex in w_{ij}, thus the optimal w_{ij}^* can be obtained by minimizing $J(\boldsymbol{\alpha}^{t+1}, w_{ij})$, or equivalently optimizing (8) or (10). Now set

$w_{ij}{}^{t+1} = w_{ij}{}^{*}$, then $J(\boldsymbol{\alpha}^{t+1}, w_{ij}{}^{t+1}) = J(\boldsymbol{\alpha}^{t+1}, w_{ij}{}^{*}) \le J(\boldsymbol{\alpha}^{t+1}, w_{ij}{}^{t})$. Finally, $J(\boldsymbol{\alpha}^{t+1}, w_{ij}{}^{t+1}) \le J(\boldsymbol{\alpha}^{t+1}, w_{ij}{}^{t}) \le J(\boldsymbol{\alpha}^{t}, w_{ij}{}^{t})$, $\forall t \in N$. Hence, the consequence $\{J(\boldsymbol{\alpha}^{t}, w_{ij}{}^{t})\}$ decreases monotonically.

Further, since the objective function is non-negative, thus lower-bounded, as a result, the sequence $\{J(\boldsymbol{\alpha}^{t}, w_{ij}{}^{t})\}$ converges.

4. Experiments

In this section, we will evaluate the performance of the AGMR over 15 UCI and benchmark datasets compared with MR, and its improvements adopting different graph construction strategies. In MR, we should construct the manifold graph before classification. In the graph construction of MR, we usually use the k-nearest neighbor and heat kernel weighting strategies. In this way, we actually preserve the local structure of the instance distribution over the manifold graph, and we will call it local preserving MR (MR_LP for short) hereafter. Further, we also construct a sparse graph by preserving the sparse structure of instance distribution, i.e., we can first get the graph weights in terms of the sparse representation,

$$\begin{aligned} &\min_{w_{i.}} \|w_{i.}\|_1 \\ &s.t. \|x_i - X w_{i.}\| < \varepsilon \\ &1 = 1^T w_{i.} \end{aligned} \tag{14}$$

where ε is the error tolerance and generally fixed across various instances of the problem. Using the graph weights from (14), we can develop a sparse preserving MR method (MR_SP for short). In the following, we will compare the AGMRs with both MR_LP and MR_SP. For solving the optimization problem in (14), we resort to the SLEP toolbox [18].

Each UCI data set is randomly split into two halves, one half for training and the other for testing, and the training set contains only 10 labeled instances and the rests are unlabeled. This process along with the classifier learning is repeated 30 times and the average testing accuracies are reported. For each benchmark datasets, there are two settings, one including 10 labeled instances and the other including 100 instances. Further, for each dataset and each setting, there are 12 subsets of labeled data, finally, the average performances on the unlabeled data are reported.

The linear kernel is adopted here. For MR_LP, the neighbor number k in the manifold graph construction is fixed to 10. When 10 instances are labeled, γ_1 and γ_2 in all compared methods are both fixed to 1, η, η_1 and η_2 in AGMR are fixed to 1, 1 and 1, respectively, ε is set to the average distance between all instance pairs. When 100 instances are labeled, the best performances over the parameter combinations from [0.01, 0.1, 1, 10, 100] are reported. The training process is repeated 20 times and the average accuracy and variance are reported in Tables 1 and 2, respectively, in which the best performance over each dataset is highlighted in bold in each row. The values with "*" over AGMR_entropy and AGMR_sparse indicate a significant improvement over MR_LP and MR_SP, respectively, through the t-test with the confidence interval at 95%.

From those tables, we can get several observations as follows:

- The performance of MR_SP is slightly better than that of MR_LP. Specifically, when 10 instances are labeled, MR_SP performs better than MR_LP on 9 out of the 15 datasets, and worse on 5 ones. When 100 instances are labeled, MR_SP performs better than MR_LP over 10 out of the 15 datasets, and worse over 5 ones. The reason can be that the manifold graph in MR_LP is constructed by the k-nearest neighbor and heat kernel weighting strategies. Those strategies turn out to be a kind of artificial sparse assumptions, which are not informative about the geodesic distances [2].
- The performance of AGMR_entropy is better than that of MR_LP. Specifically, when 10 instances are labeled, AGMR_entropy performs better than MR_LP over 9 out of the 15 datasets, with significant improvements over 4 ones, and worse over 5 ones. When 100 instances are labeled, AGMR_entropy performs better than MR_LP over 11 datasets, with significant improvements over 8 ones, and worse over 2. As a result, it is better to optimize the graph in classification rather than pre-define it.
- At the same time, the performance of AGMR_sparse is better than that of MR_SP. Specifically, when 10 instances are labeled, AGMR_sparse performs better than MR_SP over 12 out of the 15 datasets, with significant improvements over 7 ones, and worse over 4 ones. When 100 instances are labeled, AGMR_sparse per-

Algorithm 1 The algorithm description of AGMR.

Input X— the input data
γ_1, γ_2 *and* η(or η_1 and η_2) — the regularization parameter
ε — the iterative stop parameter
σ — the kernel parameter
Maxiter — the maximum number for iteration
Output $f(x)$ — the decision function
w_{ij} — the weights for the manifold graph
Procedure
1. Obtain the initial $f(x)$ by MR;
2. Set the initial objective function value to infinity, i.e., $J_0 = \mathrm{INF}$;
3. For $k = 1 \ldots$Maxiter
  Update w_{ij} by (9) or (13);
  Update $\boldsymbol{\alpha}$ by (7), and $f(x)$ by the Representer theorem with obtained $\boldsymbol{\alpha}$;
  Update the objective function vaule J^k;
  If $|J^k - J^{k-1}| < \varepsilon J^{k-1}$
    Break, return $f(x)$ and w_{ij};
  Endif
Endfor

Table 1
The comparative results with 10 labeled instances.

Dataset	MR_LP	MR_SP	AGMR_entropy	AGMR_sparse
Automobile	**79.60 ± 0.97**	78.80 ± 0.92	79.26 ± 0.82	77.32 ± 0.80
Bupa	**55.34 ± 0.15**	45.79 ± 0.18	54.88 ± 0.08	45.55 ± 0.26
Hepatitis	58.97 ± 4.05	62.00 ± 0.35	68.00 ± 0.15*	**71.86 ± 0.34***
House	53.06 ± 2.45	60.49 ± 1.10	57.25 ± 0.24*	**69.01 ± 1.10***
Ionosphere	70.29 ± 0.17	**76.77 ± 0.04**	69.71 ± 0.02	74.13 ± 2.96
Sonar	47.63 ± 0.01	47.97 ± 0.01	47.89 ± 0.02	**57.07 ± 0.01***
Spectf	54.98 ± 0.43	54.40 ± 0.73	**56.93 ± 0.33***	55.18 ± 1.07
Water	67.74 ± 0.25	75.85 ± 2.42	68.89 ± 0.14*	**76.60 ± 2.12**
Wdbc	67.30 ± 0.00	72.61 ± 0.77	68.24 ± 0.98	**74.15 ± 1.01***
Bci	50.10 ± 0.01	50.95 ± 0.14	52.25 ± 0.08*	**52.40 ± 0.18***
Ethn	59.64 ± 0.28	59.64 ± 0.78	60.10 ± 0.65	**60.65 ± 0.82***
German	61.26 ± 0.28	59.36 ± 0.53	**61.82 ± 0.31**	60.02 ± 0.53
Digit1	**56.97 ± 1.36**	54.92 ± 0.79	54.61 ± 0.10	55.15 ± 0.78
Isolet	54.21 ± 1.50	**65.43 ± 2.62**	52.89 ± 0.16	63.98 ± 0.35
Usps	57.75 ± 0.16	59.70 ± 0.03	60.40 ± 0.07*	**62.56 ± 0.07***

Table 2
The comparative results with 100 labeled instances.

Dataset	MR_LP	MR_SP	AGMR_entropy	AGMR_sparse
Automobile	89.83 ± 2.87	90.96 ± 0.96	91.13 ± 0.76*	**92.53 ± 0.89***
Bupa	63.80 ± 6.45	65.76 ± 7.86	64.98 ± 4.63*	**67.32 ± 6.54***
Hepatitis	80.20 ± 4.30	82.86 ± 1.66	82.32 ± 1.26*	**84.51 ± 1.12***
House	91.43 ± 2.31	91.97 ± 1.89	92.03 ± 1.66	**92.89 ± 1.37***
Ionosphere	88.00 ± 0.01	88.35 ± 0.01	88.50 ± 0.00	**88.98 ± 0.00**
Sonar	75.69 ± 8.51	76.85 ± 8.82	77.20 ± 8.56*	**78.59 ± 7.69***
Spectf	73.17 ± 9.71	79.79 ± 8.51	74.80 ± 9.34*	**80.04 ± 8.62**
Water	**1 ± 0**	99.22 ± 4.88	**1 ± 0**	**1 ± 0**
Wdbc	91.34 ± 6.23	89.74 ± 5.41	**91.87 ± 4.98**	90.13 ± 5.07
Bci	58.95 ± 0.10	59.40 ± 0.09	62.70 ± 0.18*	**63.54 ± 0.12***
Ethn	90.90 ± 0.58	88.41 ± 0.67	89.98 ± 6.45	**91.15 ± 6.45***
German	63.36 ± 0.10	65.94 ± 6.45	67.56 ± 6.45*	**69.54 ± 6.45***
Digit1	58.00 ± 0.12	59.20 ± 0.04	61.15 ± 0.09*	**62.38 ± 0.10***
Isolet	**98.90 ± 0.01**	97.40 ± 0.03	95.40 ± 0.07	95.00 ± 0.09
Usps	**80.00 ± 0.00**	79.85 ± 0.05	**80.00 ± 0.00**	79.99 ± 0.04

forms better than MR_SP over 13 datasets, with significant improvements over 9 ones, and worse over just 1 one. As a result, the graph optimization in classification can help boost the classification performance.

- The performance of AGMR_sparse is better than AGMR_entropy. Specifically, when 10 instances are labeled, AGMR_sparse performs better than AGMR_entropy over 11 datasets, and worse over 4 ones. When 100 instances are labeled, AGMR_sparse performs better than AGMR_entropy over 12 datasets, and worse over 3 ones. Actually, entropy maximization tends to find a uniform distribution. While sparseness seeks for a biased or sparse distribution. It introduces a trade-off between uniformity and sparseness in terms of the underlying data structure. Such a balance is data-dependent, and seeking for such a balance is a worth studying future work for us.

5. Conclusion

The performance of MR largely depends on the manifold graph, which is usually pre-constructed before the classification process. However, independently of the classification process, the constructed graph does not necessarily benefit the subsequent classification learning. At the same time, the parameters in the manifold graph of MR are difficult to set due to the limited label information in semi-supervised classification. To address those issues, we combine the graph construction and classification learning to develop a unified framework AGMR. By adopting the entropy and sparse constraints for the graph weights, respectively, we derived two specific methods called AGMR_ENTROPY and AGMR_SPARSE, respectively. Finally, empirical results show the competitiveness of AGMR compared to MR and its variants.

However, there are still works needing investigating in the future, e.g., studying the weights constraints in AGMR, and the selection of parameters in AGMR.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant No.s 61300165 and 61375057, the Specialized Research Fund for the Doctoral Program of Higher Education of Chinaunder grant No. 20133223120009, the Introduction of Talent Research Foundation of Nanjing University of Posts and Telecommunications under grand No.s NY213033.

Appendix A

Using the Lagrange multiplier method, we have

$$L = \sum_{i,j=1}^{n} w_{ij}\left(f(x_i) - f(x_j)\right)^2 + \eta \sum_{i,j=1}^{n} w_{ij} \ln w_{ij} - \sum_{i=1}^{u} \lambda_i \left(\sum_{j=1}^{u} w_{ij} - 1\right) \tag{15}$$

The derivative of L w.r.t. each w_{ij} vanishes at the minimizer, $\forall i = 1...n,\ j = 1...n$, i.e.,

$$\frac{\partial L}{\partial w_{ij}} = \left(f(x_i) - f(x_j)\right)^2 + \eta(1 + \ln w_{ij}) - \lambda_i = 0 \tag{16}$$

Thus

$$w_{ij} = e^{\frac{\lambda_i - \left(f(x_i) - f(x_j)\right)^2 - \eta}{\eta}} \tag{17}$$

Further, $\sum_{j=1}^{u} w_{ij} = 1$, thus

$$e^{\frac{\lambda_i - \eta}{\eta}} = \sum_{j=1}^{n} e^{\left(f(x_i) - f(x_j)\right)^2/\eta} \tag{18}$$

Finally,

$$w_{ij} = \frac{e^{-\left(f(x_i) - f(x_j)\right)^2/\eta}}{\sum_{j=1}^{n} e^{-\left(f(x_i) - f(x_j)\right)^2/\eta}} \tag{19}$$

Appendix B

For the solution of (12), we will use the strategy of an auxiliary function. The definition of the auxiliary function [20], a lemma in [16] and several propositions in [22] are quoted as follows:

Definition 1. $G(F, F')$ is an auxiliary function for $F(F)$ if the conditions

$$G(F, F') \geq F(F), G(F, F) = F(F)$$

are satisfied.

The auxiliary function is a useful concept because of the following lemma.

Lemma 1. If $G(F, F')$ is an auxiliary function of $F(F)$, then $F(F)$ is non-increasing under the update

$$F^{(t+1)} = \arg_F \min G(F, F') \tag{20}$$

Proposition 2. For any matrices $\mathbf{A} \in R_+^{r\times r}$, $\mathbf{W} \in R_+^{m\times r}$, and $\mathbf{W}' \in R_+^{m\times r}$, it holds

$$Tr(\mathbf{W}'^T\mathbf{W}'\mathbf{A}) \leq \sum_{ij} \frac{(\mathbf{WA})_{ij}\mathbf{W}'^2_{ij}}{\mathbf{W}_{ij}} \tag{21}$$

Proposition 3. For any matrices $\mathbf{A} \in R_+^{m\times r}$, $\mathbf{W} \in R_+^{m\times r}$, and $\mathbf{W}' \in R_+^{m\times r}$, we have

$$Tr(\mathbf{A}^T\mathbf{W}') \leq \sum_{ij} \mathbf{A}_{ij}\left(\frac{\mathbf{W}'^2_{ij} + \mathbf{W}^2_{ij}}{2\mathbf{W}_{ij}}\right) \tag{22}$$

The optimization problem in (12) can be rewritten as

$$L(\mathbf{W}) = Tr\left(\mathbf{F}^T(\mathbf{D} - \mathbf{W})\mathbf{F}\right) + \eta_1 Tr(\mathbf{X}^T\mathbf{X} - 2\mathbf{X}^T\mathbf{XW} + \mathbf{W}^T\mathbf{X}^T\mathbf{XW}) + \eta_2\|\mathbf{W}\|_1 \tag{23}$$

Lemma 2. Let the function $G(\mathbf{W}, \mathbf{W}')$ be defined as

$$G(\mathbf{W}, \mathbf{W}') = Tr(\eta_1\mathbf{X}^T\mathbf{X} - 2\eta_1\mathbf{X}^T\mathbf{XW} - \mathbf{F}^T\mathbf{WF}) + \eta_1 \sum_{ij} \frac{(\mathbf{X}^T\mathbf{XW})_{ij}\mathbf{W}'^2_{ij}}{\mathbf{W}_{ij}} + \eta_2 \sum_{ij}\left(\frac{\mathbf{W}^2_{ij} + \mathbf{W}'^2_{ij}}{2\mathbf{W}_{ij}}\right) + \sum_{ij}\left(\frac{\left((\mathbf{FF}^T)_{ii} + (\mathbf{FF}^T)_{jj}\right)(\mathbf{W}^2_{ij} + \mathbf{W}'^2_{ij})}{4\mathbf{W}_{ij}}\right) \tag{24}$$

then $G(\mathbf{W}, \mathbf{W}')$ is an auxiliary function of $L(\mathbf{W})$.

By Propositions 2 and 3, it is easy to conclude that $G(\mathbf{W}, \mathbf{W}') \geq L(\mathbf{W}')$ and $G(\mathbf{W}', \mathbf{W}') = L(\mathbf{W}')$. Therefore, the function $G(\mathbf{W}, \mathbf{W}')$ is an auxiliary function of $L(\mathbf{W}')$. With the help of the auxiliary functions $G(\mathbf{W}, \mathbf{W}')$, the update rules for $\mathbf{W}$ can be derived by minimizing $G(\mathbf{W}, \mathbf{W}')$.

The update rules are derived from setting $\partial G(\mathbf{W}, \mathbf{W}')/\partial\ \mathbf{W}'_{ij}$ to zero for all $\mathbf{W}'_{ij}$. We have

$$\frac{\partial G(\mathbf{W}, \mathbf{W}')}{\partial \mathbf{W}'_{ij}} = (-2\eta_1\mathbf{X}^T\mathbf{X} - \mathbf{FF}^T)_{ij} + 2\eta_1\frac{(\mathbf{X}^T\mathbf{XW})_{ij}\mathbf{W}'_{ij}}{\mathbf{W}_{ij}} + \eta_2\left(\frac{\mathbf{W}'_{ij}}{\mathbf{W}_{ij}}\right) + \frac{\left((\mathbf{FF}^T)_{ii} + (\mathbf{FF}^T)_{jj}\right)\mathbf{W}'_{ij}}{2\mathbf{W}_{ij}} = 0 \tag{25}$$

Finally, the updated rule can be formulated as

$$w_{ki} = w_{ki}\frac{\left(\eta_1\mathbf{X}^T\mathbf{X} + \frac{1}{2}\mathbf{FF}^T\right)_{ki}}{(\eta_1\mathbf{X}^T\mathbf{XW})_{ki} + \frac{1}{4}\left(\left(\mathbf{FF}^T\right)_{ii} + \left(\mathbf{FF}^T\right)_{kk}\right) + \frac{\eta_2}{2}} \tag{26}$$

where $\mathbf{X}$ is the data matrix, i.e., $\mathbf{X} = [x_1, x_2, ...x_n] \in R^{d\times n}$, where $x_i \in R^d$ and n is the total number of instances. $\mathbf{F}$ is the column vector of classification scores for instances, i.e., $\mathbf{F} = [f(x_1), ..., f(x_n)]' \in R^n$, where $f(x_i)$ is the classification score for each instance x_i.

References

[1] J. Abernethy, O. Chapelle, C. Castillo, Web spam identification through content and hyperlinks, in: Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web, Beijing, China, 2008, pp. 41–44.

[2] M. Alamgir, U.V. Luxburg, Shortest Path Distance in Random K-Nearest Neighbor Graphs, Computer Science, 2012.
[3] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J.Mach. Learn. Res 7 (2006) 2399–2434.
[4] Y. Bengio, O.B. Alleau, N.L. Roux, et al., Label propagation and quadratic criterion, in: O. Chapelle, et al. (Eds.), Semi-Supervised Learning, MIT Press, 2006, pp. 193–216.
[5] A. Blum, S. Chawla, Learning from labeled and unlabeled data using graph mincuts, in: Proceedings of the 18th International Conference on Machine Learning, Williamstown, MA, USA, 2001, pp. 19–26.
[6] T.M. Cover, J.A. Thomas, Entropy, relative entropy, and mutual information, in: Elements of Information Theory, second ed., John Wiley & Sons, 2005, pp. 13–55.
[7] O. Chapelle, B. Scholkopf, A. Zien, Semi-Supervised Learning, MIT Press, 2006.
[8] R. Collobert, F. Sinz, J. Weston, L. Bottou, Large scale transductive SVMs, J. Mach. Learn.Res. 7 (2006) 1687–1712.
[9] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J.R. Stat. Soc. Ser. B 39 (1977) 1–38.
[10] F. Escolano, P. Suau, B. Bonev, Information Theory in Computer Vision and Pattern Recognition, Springer, London, 2009.
[11] G. Fung, O.L. Mangasarian, Semi-supervised support vector machine for unlabeled data classification, Optim. Meth. Soft. 15 (2001) 99–105.
[12] B. Geng, D. Tao, C. Xu, L. Yang, X.-S. Hua, Ensemble manifold regularization, IEEE Trans.Pattern Anal. Mach. Intell. 34 (2012) 1227–1233.
[13] J. Gorski, F. Pfeuffer, Biconvex sets and optimization with biconvex functions: a survey and extensions, Math. Meth. Oper. Res. 66 (2007) 373–407.
[14] X. He, Laplacian regularized D-optimal design for active learning and its application to image retrieval, IEEE Trans. Image Process 19 (2010) 254–263.
[15] T. Joachims, Transductive inference for text classification using support vector machines, in: Proceedings of the 16th International Conference on Machine Learning, Bled, Slovenia, 1999, pp. 200–209.
[16] D.D. Lee, H.S. Seung, Learning the parts of objects by nonnegative matrix factorization, Nature 401 (1999) 788–791.
[17] Y.-F. Li, J.T. Kwok, Z.-H. Zhou, Semi-supervised learning using label mean, in: Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada, 2009, pp. 633–640.
[18] J. Liu, S. Ji, J. Ye, SLEP: sparse learning with efficient projections, Arizona State University, 2010.
[19] P.K. Mallapragada, R. Jin, A.K. Jain, Y. Liu, Semi-boost: boosting for semi-supervised learning, IEEE Trans. Pattern Anal.Mach. Intell. 31 (2009) 2000–2014.
[20] X. Pei, Z. Lyu, C. Chen, C. Chen, Manifold adaptive label propagation for face clustering, IEEE Trans.Cybern. 45 (2015) 1681–1691.
[21] I.W. Tsang, J.T. Kwok, Large-scale sparsified manifold regularization, Advances in Neural Information Processing Systems, Vancouver, B.C., Canada, 2006.
[22] Z. Yang, E. Oja, Linear and nonlinear projective nonnegative matrix factorization, IEEE Trans. Neural Netw. Learn. Syst. 21 (2010) 734–749.
[23] K. Zhang, J.T. Kwok, B. Parvin, Prototype vector machine for large scale semi–supervised learning, in: Proceedings of the 26 th International Conference on Machine Learning, Montreal, Canada, 2009, pp. 1233–1240.
[24] L. Zhang, L. Qiao, S. Chen, Graph-optimized locality preserving projections, Pattern Recognit. 43 (2010) 1993–2002.
[25] Z.-H. Zhou, M. Li, Semi-supervised learning by disagreement, Knowl. Inf. Syst. 24 (2010) 415–439.
[26] X. Zhu, Semi-supervised learning literature survey, Computer Sciences, University of Wisconsin-Madison, 2008.
[27] X. Zhu, Z. Ghahramani, Learning From Labeled And Unlabeled Data With Label Propagation, Carnegie Mellon University, 2002.
[28] X. Zhu, A.B. Goldberg, Introduction to Semi-Supervised Learning, Morgan & Claypool, San Rafael, 2009.