Int J Software Informatics, Volume 7, Issue 3 (2013), pp.419–433 Tel: +86-10-62661040
International Journal of Software and Informatics, ISSN 1673-7288 http://www.ijsi.org

Email: ijsi@iscas.ac.cn, ijsi2007@gmail.com

Bidirectional Label Propagation over Graphs

Wei Liu[1] and Tongtao Zhang[2]

[1] (IBM T. J. Watson Research Center, Yorktown Heights, NY, USA)
[2] (Columbia University, New York, NY, USA)

Abstract Graph-Based label propagation algorithms are popular in the state-of-the-art semi-supervised learning research. The key idea underlying this algorithmic family is to enforce labeling consistency between any two examples with a positive similarity. However, negative similarities or dissimilarities are equivalently valuable in practice. To this end, we simultaneously leverage similarities and dissimilarities in our proposed semi-supervised learning algorithm which we term *Bidirectional Label Propagation* (BLP). Different from previous label propagation mechanisms that proceed along a single direction of graph edges, the BLP algorithm can propagate labels along not only positive but also negative edge directions. By using an initial neighborhood graph and class assignment constraints inherent among the labeled examples, a set of class-specific graphs are learned, which include both positive and negative edges and thus reveal discriminative cues. Over the learned graphs, a convex propagation criterion is carried out to ensure consistent labelings along the positive edges and inconsistent labelings along the negative edges. Experimental evidence discovered in synthetic and real-world datasets validates excellent performance of the proposed BLP algorithm.
Key words: semi-supervised learning; graph; bidirectional label propagation

Liu W, Zhang TT. Bidirectional label propagation over graphs. *Int J Software Informatics*, **Vol.7, No.3 (2013): 419–433. http://www.ijsi.org/1673-7288/7/i168.htm**

1 Introduction

In practical applications of machine learning and pattern recognition, one frequently encounters the very situation where only a few labeled examples are available for training and a great number of examples remain unlabeled. As large amounts of unlabeled data can be automatically or cheaply gathered, *Semi-Supervised Learning* (SSL)[25], an emerging important machine learning technique, is coined to deal with the situation of sparsely labeled data and abundant unlabeled data. The semi-supervised learning scenario has practical utility on many real-world problems, since it is feasible to collect unlabeled data by an automatic procedure without users' intervention but expensive for users to identify labels of data.

SSL has spurred a lot of efforts in designing effective and efficient algorithms which aim to mitigate the performance limitations of traditional supervised learning methods trained on a small set of labeled examples through leveraging a large pool of unlabeled examples. Among the recent work on semi-supervised classification, the

This work is sponsored by the Josef Raviv Memorial Postdoctoral Fellowship.
Corresponding author: Wei Liu, Email: weiliu@us.ibm.com
Received 2012-10-24; Revised 2013-08-15; Accepted 2013-08-31.

Transductive Support Vector Machine (TSVM)[10] attempted to optimize the margins of both labeled and unlabeled examples. Following TSVMs, Ref. [6] and Ref. [4] did the cluster-based inference to explore the probable decision boundary for classification that could exist in the low-density regions of the input sample space. A big family of graph-based approaches, including Refs. [2,3,12,13,19,21,24,26], founded on spectral graph theory[5], established a variety of regularization frameworks by introducing convex regularization penalties embedding *graph Laplacians*.

The paramount foundation of SSL is an appropriate assumption about the underlying data structure. Two commonly adopted assumptions are the *cluster assumption* and the *manifold assumption*. The former assumes that data samples associated with the same structure, typically a cluster or a manifold, probably take similar class or category labels[4,6]. The latter often implies that close-by sample points on the same manifold are very likely to take the same label. Note that the cluster assumption is made globally whereas the manifold assumption often holds locally. Numerous SSL methods such as the representative ones[2,19,24,26] exploited such a manifold assumption to pursue smooth prediction functions for classification or regression along manifolds. Specifically, all these methods represent both labeled and unlabeled samples into a graph, and employ the graph Laplacian matrix to discretely approximate the data manifolds. Reference [11] unified the cluster and manifold assumptions into a single optimization criterion, which actually extends the TSVM by accessing the manifold structure.

This paper is arranged as follows. Section 2 reviews the related work on graph-based semi-supervised learning. Section 3 presents the key idea of our proposed SSL method and gives an algorithmic paradigm for handling transductive and inductive learning together. Section 4 validates the effectiveness of the proposed SSL method through experiments. Section 5 includes our conclusion and discussion.

2 Related Work

Although graph-based SSL has been studied extensively, it often lacks sufficient robustness in real-world learning tasks because of the sensitivity of graphs. The quality of graphs is very sensitive to the edge connection, the choice of edge weighting functions, and the related parameters. These factors will considerably influence the performance of SSL algorithms.

In particular, the representative graph-based SSL algorithms[3,24,26], which are akin to each other, heavily depend on graphs because they only use graphs to infer the labels of unlabeled data. Moreover, they only invoke the similarities among adjacent data points, which are encoded into positive weights of graph edges. In doing so, the graph construction scheme shared by these methods is unsupervised and thereby likely to be confounded by complex multi-class data distributions. It is intuitive that negative similarities or *dissimilarities* are useful for discovering the discriminative cues hidden in multiple manifolds formed by multi-class data samples. Actually, we have known partial knowledge about dissimilarities, *i.e.*, the "*cannot-be-the-same-class*" relationship inherent among the samples from different classes.

Let us consider the classical two-moons toy problem to explain our motivation for exploiting dissimilarities. As shown in Fig. 1, we are given a set of points in a shape of two moons plus some extra points which are outliers. We simply consider these extra

points as noisy points. Two points on the upper moon and lower moon are labeled as '+1' and '−1', respectively. Intuitively, the points on the upper moon should be labeled as '+1' while those on the lower moon should be '−1'. Using the traditional $k-$NN graph ($k = 10$), we run two well-known graph-based SSL algorithms: the *Local and Global Consistency* (LGC) method[24] and the *Gaussian Fields and Harmonic Functions* (GFHF) method[26] for this toy problem. We only have ground truth labels for the points on two moons, so we evaluate classification performance over these on-manifold points. The visual classification results are displayed in Fig. 1, from which we observe quite a few errors caused by LGC and GFHF but the zero mistake accomplished by our *Bidirectional Label Propagation* (BLP) method that will be proposed in Section 3. Our BLP method succeeds in invoking both similarities and dissimilarities to perform more sophisticated label propagation than conventional similarity-driven propagation.

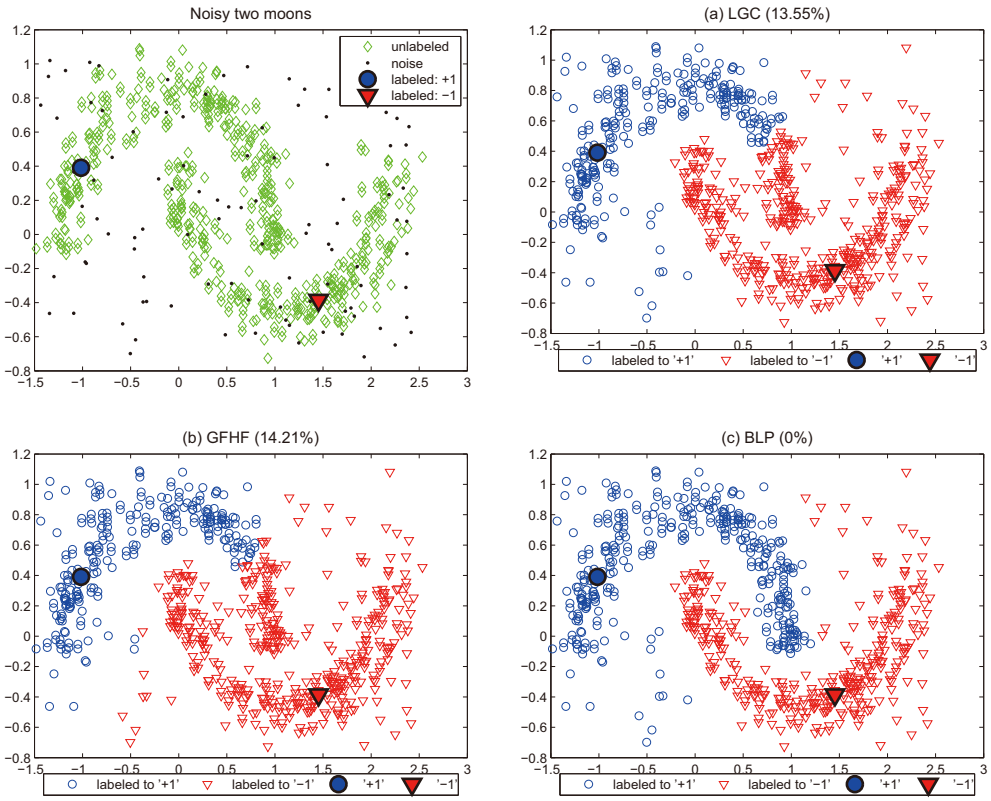

Figure 1. The noisy two-moons problem given two labeled points. (a) LGC[24] achieves 13.55% error rate with a 10-NN graph; (b) GFHF[26] achieves 14.21% error rate with a 10-NN graph; (c) our method BLP achieves zero error rate with a 10-NN graph.

The importance of dissimilarities to graph-based SSL has been realized by the recent work[8,20]. Reference [8] modified the graph Laplacian matrix by incorporating prior dissimilarity information of labeled data, and applied manifold regularization using the modified graph Laplacian. Reference [20] directly used the input dissimilarities that naturally arise from collaborative filtering problems. We

find that the mixed label propagation algorithm proposed in Ref. [20] is computationally expensive and can only be applied to the context of collaborative filtering. In this paper, we intend to infer dissimilarities among all data and develop an algorithm for general multi-class semi-supervised classification.

3 Bidirectional Label Propagation

Our SSL approach is based on a geometric intuition that for many real-world problems unlabeled data examples often reveal data structures, such as clusters or low-dimensional manifolds, which provide the useful prior knowledge and potentially help the label inference. For example, one may expect high correlations among class labels of examples within the same cluster or on the same local manifold.

This section will address the typical multi-class semi-supervised classification task. We propose to learn the class-specific graphs under the semi-supervised learning scenario and subsequently leverage such graphs into multi-class label propagation. Both graph learning and label propagation collaborate well in our proposed approach.

Suppose that there are C classes $\{\Omega^c\}_{c=1}^{C}$ appearing in the labeled data subset $\{(\boldsymbol{x}_i, y_i)|y_i \in \{1, \cdots, C\}\}_{i=1}^{l}$ that consists of l examples. If $\boldsymbol{x}_i \in \Omega^c$ then $y_i = c$. In each class Ω^c, there are $l_c = |\Omega^c|$ labeled examples.

3.1 Initial k-NN graph

Graph-based machine learning methods presume that data samples are represented in the form of undirected or directed graphs. Graph-based SSL methods frequently adopt undirected graphs. In this paper, we aim at learning a set of real-valued label prediction functions that take as input an undirected weighted graph $G = (V, E, \omega)$. V is a set of vertices with each of them representing a data sample (point), $E \subseteq V \times V$ is a set of edges each of which connects adjacent data points, and $\omega : E \to \mathbb{R}^+$ is a weighting function that measures the strength of each edge. The graph representation has been demonstrated to be effective for data which lie in compact clusters or intrinsic low-dimensional manifolds. More importantly, graphs naturally characterize pairwise proximities among data objects, which have been utilized for data clustering, embedding, visualization, and ranking. This form of data graphs has also been a central focus in computational geometry areas such as manifold learning[1].

Consider a full data set $\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_l, \cdots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$ of which, without loss of generality, the first l samples are assumed labeled and the remaining $n-l$ ones are unlabeled. In the graph G each vertex (or node) v_i corresponds to each sample $\boldsymbol{x}_i$, so we also refer to $\boldsymbol{x}_i$ as a graph node. Then we put an edge between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ if $\boldsymbol{x}_i$ is among the k nearest neighbors of $\boldsymbol{x}_j$ or $\boldsymbol{x}_j$ is among the k nearest neighbors of $\boldsymbol{x}_i$. G thus becomes a k-NN graph. Although there are other strategies for building edges over data points, it turns out that k-NN graphs have advantages over others (e.g., h-neighborhood graphs) as shown in Ref. [9]. One of main advantages is that a k-NN graph provides a better adaptive connectivity because data points in areas of different densities have different neighborhood scales while the fixed h may lead to either disconnected or over-connected graphs.

We define the weighted adjacency matrix $W \in \mathbb{R}^{n\times n}$ of G as follows

$$W_{ij} = \begin{cases} \exp\left(-\dfrac{d(\boldsymbol{x}_i, \boldsymbol{x}_j)^2}{\sigma^2}\right), & \boldsymbol{x}_j \in N(\boldsymbol{x}_i) \text{ or } \boldsymbol{x}_i \in N(\boldsymbol{x}_j) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

in which the set $N(\boldsymbol{x}_i)$ saves the k nearest neighbors of point $\boldsymbol{x}_i$ in $\mathcal{X}$ and $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ denotes some distance function between points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Typically, $d(,)$ refers to the Euclidean distance. The width parameter σ is empirically estimated by $\sigma = \sum_{i=1}^{n} d(\boldsymbol{x}_i, \boldsymbol{x}_{i_k})/n$ where $\boldsymbol{x}_{i_k}$ is defined as the k-th nearest neighbor of point $\boldsymbol{x}_i$. Such an estimation is simple and effective enough, as has been verified in Ref. [12]. Obviously, W is a symmetric matrix. Notice that we set $W_{ii} = 1$ in order to make W easily invertible.

3.2 Learning class-specific graphs

As mentioned before, the initial k-NN graph is constructed in an unsupervised manner. Since the partial labelings of the dataset $\mathcal{X}$ are available, we would prefer learning graphs in conjunct with the known label information. To well handle multi-class problems, we adopt the one-against-all strategy to convert a multi-class problem to multiple binary one-versus-rest problems. For each class Ω^c, we find two types of pairwise constraints imposed on the co-labelings:

1) the *must-link* constraint $(\boldsymbol{x}_i, \boldsymbol{x}_j)_{\mathcal{M}}$ with $y_i = y_j$, and

2) the *cannot-link* constraint $(\boldsymbol{x}_i, \boldsymbol{x}_j)_{\mathcal{C}}$ with $y_i \neq y_j$ and one of y_i, y_j being c.

For each class Ω^c we collect all class-specific constraints in a set Θ^c, where each must-link requires two labeled examples to be assigned to the same class label while each cannot-link requires two labeled examples to be assigned to the label c and a different label. Note that Θ^c provides the exact co-labeling knowledge of l labeled examples. In contrast, the adjacency matrix W of the initially constructed k-NN graph can estimate the co-labeling probabilities of all example pairs. Specially, $W_{ii} = 1$ stands for the true decision of assigning the same label for the same example.

The nature of graphs is nonparametric and the graph adjacency matrix W induces a nonparametric prior about the co-labeling decisions. Thus, we model a *Gaussian Process* (GP)[18] to define the nonparametric prior for the target label prediction functions. Suppose that a random binary label prediction function $g : \mathcal{X} \to \mathbb{R}$ is relaxed from the hard binary output $\{+1, -1\}$. Let the vectorial representation $\boldsymbol{g} \in \mathbb{R}^n$ be drawn from a GP with the zero mean and the covariance W (note that we use it as $W + 10^{-6} * I$ to guarantee the invertibility), which is sensible because $W_{ij} > 0$ implies that g_i and g_j are very likely to take on the same sign, *i.e.*, $\mathbb{E}(g_i g_j) > 0$.

Through the one-against-all strategy, we have C binary-class problems at hand, each of which corresponds to a co-labeling constraint set Θ^c since Θ^c perfectly separates the class Ω^c from the others in the labeled subset. For brevity, we denote each binary-class task as $t(\Theta^c)$, and we would expect $g_i = 1$ for $\boldsymbol{x}_i \in \Omega^c$ and $g_i = -1$ for $\boldsymbol{x}_i \notin \Omega^c$, respectively. In addition, we derive $g_i = g_j$ according to the must-link $(\boldsymbol{x}_i, \boldsymbol{x}_j)_{\mathcal{M}}$ and $g_i = -g_j$ to the cannot-link $(\boldsymbol{x}_i, \boldsymbol{x}_j)_{\mathcal{C}}$, respectively. As such, we model the likelihood of Θ^c with respect to $\boldsymbol{g} = \begin{bmatrix} \boldsymbol{g}_\ell \\ \boldsymbol{g}_\upsilon \end{bmatrix}$ ($\boldsymbol{g}_\ell \in \mathbb{R}^l$ refers to the labeled

part and $\boldsymbol{g}_v \in \mathbb{R}^{n-l}$ to the unlabeled part) as

$$\begin{aligned}\mathbf{P}(\Theta^c|\boldsymbol{g}) &\propto \exp\left(-\sum_{y_i=y_j}\frac{(g_i-g_j)^2}{2\varepsilon^2}-\sum_{\substack{y_i\neq y_j\\ y_i=c \text{ or } y_j=c}}\frac{(g_i+g_j)^2}{2\varepsilon^2}\right)\\ &= \exp\left(-\frac{1}{2}\boldsymbol{g}_\ell^\top S^c \boldsymbol{g}_\ell\right),\end{aligned} \tag{2}$$

where $S^c \in \mathbb{R}^{l\times l}$ with entries being

$$S_{ij}^c = \frac{2}{\varepsilon^2}\begin{cases} l-1, & i=j \text{ and } y_i=c,\\ l_c+l_b-1, & i=j \text{ and } y_i=b\neq c,\\ -1, & i\neq j \text{ and } y_i=y_j,\\ 1, & i\neq j \text{ and } y_i\neq y_j \text{ and } (y_i \text{ or } y_j)=c,\\ 0, & \text{otherwise.}\end{cases} \tag{3}$$

Given the GP prior $\boldsymbol{g}\sim\mathcal{N}(0,W)$, the posterior $\mathbf{P}(\boldsymbol{g}|\Theta^c)$ is derived by using the Bayes' law as follows:

$$\begin{aligned}\mathbf{P}(\boldsymbol{g}|\Theta^c) &\propto \mathbf{P}(\Theta^c|\boldsymbol{g})\mathbf{P}(\boldsymbol{g})\\ &\propto \exp\left(-\frac{1}{2}\boldsymbol{g}_\ell^\top S^c\boldsymbol{g}_\ell\right)\exp\left(-\frac{1}{2}\boldsymbol{g}^\top W^{-1}\boldsymbol{g}\right)\\ &= \exp\left(-\frac{1}{2}\boldsymbol{g}^\top (K^c)^{-1}\boldsymbol{g}\right),\end{aligned} \tag{4}$$

where $K^c \in \mathbb{R}^{n\times n}$ is the covariance of the derived posterior process $\boldsymbol{g}|\Theta^c$ which is still a GP with the zero mean. The following theorem gives the exact form of K^c.

Theorem 3.1. *If we write W in the blockwise form $\begin{bmatrix} W_{\ell\ell} & W_{\ell v}\\ W_{v\ell} & W_{vv}\end{bmatrix}$ according to the partition of the labeled and unlabeled examples, then*

$$K^c = \begin{bmatrix} W_{\ell\ell}-W_{\ell\ell}T^cW_{\ell\ell} & W_{\ell v}-W_{\ell\ell}T^cW_{\ell v}\\ W_{v\ell}-W_{v\ell}T^cW_{\ell\ell} & W_{vv}-W_{v\ell}T^cW_{\ell v}\end{bmatrix}, \tag{5}$$

where $T^c=(I+S^cW_{\ell\ell})^{-1}S^c\in\mathbb{R}^{l\times l}$.

proof: Taking Eq. (4) into account, we can deduce K^c by applying matrix inversion lemma and block matrix inversion, that is,

$$\begin{aligned}K^c &= \left(\begin{bmatrix} S^c & 0\\ 0 & 0\end{bmatrix}+W^{-1}\right)^{-1}\\ &= W - W\left(I+\begin{bmatrix} S^c & 0\\ 0 & 0\end{bmatrix}W\right)^{-1}\begin{bmatrix} S^c & 0\\ 0 & 0\end{bmatrix}W\\ &= W - W\begin{bmatrix} I+S^cW_{\ell\ell} & S^cW_{\ell v}\\ 0 & I\end{bmatrix}^{-1}\begin{bmatrix} S^cW_{\ell\ell} & S^cW_{\ell v}\\ 0 & 0\end{bmatrix}\end{aligned}$$

$$= W - W \begin{bmatrix} (I + S^c W_{\ell\ell})^{-1} S^c W_{\ell\ell} & (I + S^c W_{\ell\ell})^{-1} S^c W_{\ell v} \\ 0 & 0 \end{bmatrix}$$
$$= W - W \begin{bmatrix} T^c W_{\ell\ell} & T^c W_{\ell v} \\ 0 & 0 \end{bmatrix},$$

which immediately leads to Eq. (5). □

We exploit this covariance K^c of the derived posterior GP $\boldsymbol{g}|\Theta^c$ as the class-specific affinity (*i.e.*, adjacency) matrix for dealing with the task $t(\Theta^c)$. It is not difficult to prove the positive definiteness of the matrix K^c, so K^c can be regarded as a valid kernel matrix. Different from the initial affinity matrix W, the class-specific affinity matrix K^c includes both positive and negative similarities, which are visualized by the illustrative example in Fig. 2. Naturally, the learned affinity matrix K^c specifies a novel graph $\mathcal{G}^c$ that is also class-specific. We call the edges in $\mathcal{G}^c$ taking positive similarities as *positive edges*, and the edges taking negative similarities (*i.e.*, dissimilarities) as *negative edges*.

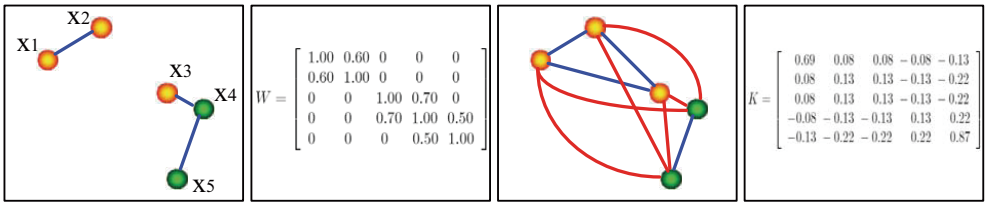

Figure 2. An example of learning class-specific graphs. The left two subfigures show the initial 1-NN graph and the associated affinity matrix W. Given a must-link between x_2 and x_3 and a cannot-link between x_3 and x_4, the right two subfigures show the learned class-specific affinity matrix K which specifies a novel graph of positive and negative edges. Different colors for data points represent different class labels. Blue edges (positive edges) take similarities (positive edge weights), while red edges (negative edges) take dissimilarities (negative edge weights).

Let us revisit Eq. (5), in which the matrix S^c encodes the co-labeling constraints of the set Θ^c imposed on the labeled block, and then diffuses the co-labeling cues to the rest of the blocks of the initial affinity matrix W. Therefore, K^c may be thought of as the response of W after absorbing the co-labeling cues offered by Θ^c. We simply set the parameter ε in S^c, displayed in Eq. (3), to a very small value such as 10^{-3} so as to make the co-labeling constraints as hard as possible. When more than two classes are confronted, we have to learn C affinity matrices. When $C = 2$, we only need to learn a single affinity matrix.

3.3 Bidirectional label propagation criterion

Over the learned class-specific graph $\mathcal{G}^c(K^c)$, we propose a *Bidirectional Label Propagation* (BLP) criterion that takes advantage of both similarities and dissimilarities, embedded into K^c, among the given n examples:

$$\mathcal{Q}(\boldsymbol{f}_c) = \frac{1}{2} \sum_{K^c_{ij} \geqslant 0} (f_{ic} - f_{jc})^2 K^c_{ij} - \frac{1}{2} \sum_{K^c_{ij} < 0} (f_{ic} + f_{jc})^2 K^c_{ij}$$

$$= \text{Incons}^{+}(\boldsymbol{f}_c) + \text{Cons}^{-}(\boldsymbol{f}_c), \tag{6}$$

where $\boldsymbol{f}_c = [f_{1c}, \cdots, f_{nc}]^\top \in \mathbb{R}^n$ saves the predicted soft labels of the examples in $\mathcal{X}$ for the task $t(\Theta^c)$, Incons^{+} measures the labeling inconsistency along the positive edges of the graph $\mathcal{G}^c$, and Cons^{-} measures the labeling consistency along the negative edges. Clearly, $\mathcal{Q}(\boldsymbol{f}_c) \geqslant 0$ always holds for arbitrary $\boldsymbol{f}_c$. The optimal $\boldsymbol{f}_c$ must minimize the BLP criterion in Eq. (6).

Let us compute the $n \times n$ matrix

$$B^c = D^c - K^c, \tag{7}$$

where $D^c \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose diagonal entries are $D^c_{ii} = \sum_{j=1}^{n} |K^c_{ij}|$.

Theorem 3.2. *The proposed bidirectional label propagation criterion constitutes a convex quadratic function*

$$\mathcal{Q}(\boldsymbol{f}_c) = \boldsymbol{f}_c^\top B^c \boldsymbol{f}_c, \tag{8}$$

where B^c is positive semidefinite.

proof: We deduce $\mathcal{Q}(\boldsymbol{f}_c)$ as follows

$$\begin{aligned}
\mathcal{Q}(\boldsymbol{f}_c) &= \sum_{i=1}^{n} f_{ic}^2 \sum_{j, K^c_{ij} \geqslant 0} K^c_{ij} - \sum_{K^c_{ij} \geqslant 0} f_{ic} f_{jc} K^c_{ij} \\
&\quad - \sum_{i=1}^{n} f_{ic}^2 \sum_{j, K^c_{ij} < 0} K^c_{ij} - \sum_{K^c_{ij} < 0} f_{ic} f_{jc} K^c_{ij} \\
&= \sum_{i=1}^{n} f_{ic}^2 \sum_{j=1}^{n} |K^c_{ij}| - \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ic} f_{jc} K^c_{ij} \\
&= \boldsymbol{f}_c^\top \left(D^c - K^c\right) \boldsymbol{f}_c \\
&= \boldsymbol{f}_c^\top B^c \boldsymbol{f}_c,
\end{aligned}$$

which indicates that $\mathcal{Q}(\boldsymbol{f}_c)$ is quadratic in terms of $\boldsymbol{f}_c$. Additionally, because $\mathcal{Q}(\boldsymbol{f}_c) \geqslant 0$ for any $\boldsymbol{f}_c$, $\mathcal{Q}$ is convex and accordingly B^c is a positive semidefinite matrix. □

By utilizing Eq. (8), we can establish the following constrained regularization framework to execute bidirectional label propagation for the SSL task $t(\Theta^c)$:

$$\begin{aligned}
\min_{\boldsymbol{f}_c} \quad & \boldsymbol{f}_c^\top B^c \boldsymbol{f}_c + \xi \|\boldsymbol{f}_c\|^2 \\
\text{s.t.} \quad & \boldsymbol{f}_{\ell,c} = Y_{\cdot c}
\end{aligned} \tag{9}$$

in which $\boldsymbol{f}_c = \begin{bmatrix} \boldsymbol{f}_{\ell,c} \\ \boldsymbol{f}_{\upsilon,c} \end{bmatrix}$, $Y_{\cdot c} = \begin{bmatrix} Y_{1c} \\ \cdots \\ Y_{lc} \end{bmatrix} \in \mathbb{R}^l$, and $\xi > 0$ is the regularization parameter. We predefine $Y_{ic} = 1$ if $y_i = c$ and $Y_{ic} = -1$ otherwise. With simple algebra, Eq. (9) reduces to

$$\min_{\boldsymbol{f}_{\upsilon,c}} \quad \mathcal{Q}_1(\boldsymbol{f}_{\upsilon,c}) = \boldsymbol{f}_{\upsilon,c}^\top (B^c_{\upsilon\upsilon} + \xi I) \boldsymbol{f}_{\upsilon,c} + 2 \boldsymbol{f}_{\upsilon,c}^\top B^c_{\upsilon\ell} Y_{\cdot c}, \tag{10}$$

where B^c_{vv} and $B^c_{v\ell}$ are sub-matrices of $B^c = \begin{bmatrix} B^c_{\ell\ell} & B^c_{\ell v} \\ B^c_{v\ell} & B^c_{vv} \end{bmatrix}$. We let $\partial \mathcal{Q}_1 / \partial \boldsymbol{f}_{v,c} = 0$ and then obtain the globally optimal solution to Eq. (9) as follows

$$\boldsymbol{f}^*_{v,c} = -(B^c_{vv} + \xi I)^{-1} B^c_{v\ell} Y_{\cdot c}. \tag{11}$$

3.4 Inductive inference

It is worthwhile to state that truly semi-supervised learning, *e.g.*, Ref. [19] and Ref. [2], should handle training examples in availability as well as unseen test examples. Generally speaking, transductive learning such as Refs. [24,26] can only infer the labels of the training examples and fails to infer the labels of any novel examples beyond the training dataset. Reference [7] followed the same label propagation criteria presented in Refs. [24,26] and developed a nonparametric *inductive inference* scheme for predicting the labels of any out-of-sample examples.

In this paper, we desire to investigate an inductive inference scheme which can yield an out-of-sample extension of the proposed BLP criterion. Specifically, we not only conduct induction for the label $y(\mathbf{z})$ of a novel example $\boldsymbol{z} \in \mathbb{R}^d$ but also perform induction for the affinities $K^c(\boldsymbol{z}, \boldsymbol{x}_i) = K^c_{zi}$ between the novel example $\boldsymbol{z}$ and n existing training examples $\{\boldsymbol{x}_i\}_{i=1}^n$.

Above all, we define the initial affinity $W(\boldsymbol{z}, \boldsymbol{x}_i)$ that we rewrite as W_{zi} for brevity:

$$W_{zi} = \begin{cases} \exp\left(-\frac{d(\boldsymbol{z},\boldsymbol{x}_i)^2}{\sigma^2}\right), & \boldsymbol{x}_i \in N(\boldsymbol{z}) \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

We also set $W_{zz} = 1$ in accordance with $W_{ii} = 1$.

Theorem 3.3 *Let* $W_{z\cdot} = [W_{z1}, \cdots, W_{zl}, \cdots, W_{zn}] = [W_{z,\ell}, W_{z,v}]$, *then*

$$\begin{aligned} K^c_{z\cdot} &= [K^c_{z1}, \cdots, K^c_{zl}, \cdots, K^c_{zn}] \\ &= W_{z\cdot} - W_{z,\ell} T^c W_{\ell\cdot}, \end{aligned} \tag{13}$$

where $W_{\ell\cdot} = [W_{\ell\ell}, W_{\ell v}] \in \mathbb{R}^{l \times n}$.

proof: We can assume $\boldsymbol{z} = \boldsymbol{x}_{n+1}$ and derive the same form as Eq. (5). Let substitute $\begin{bmatrix} K^c & K^{c\top}_{z\cdot} \\ K^c_{z\cdot} & K^c_{zz} \end{bmatrix}$ for K^c, $\begin{bmatrix} W_{v\ell} \\ W_{z,\ell} \end{bmatrix}$ for $W_{v\ell}$, and $\begin{bmatrix} W_{vv} & W^\top_{z,v} \\ W_{z,v} & 1 \end{bmatrix}$ for W_{vv} in Eq. (5), respectively. Then we can obtain $K^c_{z\cdot}$ as formulated in Eq. (13) by equating two sides of Eq. (5). □

So far, we can apply Theorem 3.3 and the same BLP criterion in the transductive learning setting to infer the label of any out-of-sample example $\boldsymbol{z}$. Specifically, we suppose a pseudo label f_{zc} for $\boldsymbol{z}$ in the task $t(\Theta^c)$, which is solved through minimizing the following cost function

$$\mathcal{Q}_2(f_{zc}) = \frac{1}{2} \sum_{K^c_{zj} \geqslant 0} (f_{zc} - f_{jc})^2 K^c_{zj} - \frac{1}{2} \sum_{K^c_{zj} < 0} (f_{zc} + f_{jc})^2 K^c_{zj}. \tag{14}$$

Let $\partial \mathcal{Q}_2 / \partial f_{zc} = 0$, $f_{jc} = Y_{jc}$ for $1 \leqslant j \leqslant l$, and $f_{jc} = f^*_{jc}$ for $j \geqslant l+1$. We thus

Algorithm 1 Bidirectional Label Propagation (BLP)

Step 1: Construct a k-NN graph $G(V, E, W)$ upon $\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_l, \cdots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$. The first l examples are labeled as $y_i \in \{1, \cdots, C\}$. Use Eq. (1) to set W. Set a class indicator matrix $Y \in \mathbb{R}^{l \times C}$ where $Y_{ic} = 1$ if $y_i = c$ and $Y_{ic} = -1$ otherwise.
Step 2:
Transductive Inference:
for $c = 1, 2, \cdots, C$ **do**
use eqs. (3)(5) to compute K^c from W,
apply Eq. (11) to compute $\boldsymbol{f}^*_{v,c}$ with K^c and $Y_{\cdot c}$,
.
$F^*_v = [\boldsymbol{f}^*_{v,1}, \cdots, \boldsymbol{f}^*_{v,C}]$.
Inductive Inference:
for $c = 1, 2, \cdots, C$ **do**
use eqs. (12)(13) to compute $K^c_{z\cdot}$,
apply Eq. (15) to compute f^*_{zc} with $K^c_{z\cdot}$, $Y_{\cdot c}$, and $\boldsymbol{f}^*_{v,c}$,
.
$F^*_z = [f^*_{z1}, \cdots, f^*_{zC}]$.
Step 3: For an unlabeled example $\boldsymbol{x}_i$ $(l+1 \leqslant i \leqslant n)$, predict its labels by $y(\boldsymbol{x}_i) = \arg\max_{1 \leqslant c \leqslant C} [F^*_u]_{i-l,c}$. For a novel test example $\boldsymbol{z}$, predict its label by $y(\boldsymbol{z}) = \arg\max_{1 \leqslant c \leqslant C} [F^*_z]_{1,c}$.

accomplish the optimal solution to Eq. (14) as follows

$$f^*_{zc} = \frac{\sum_{j=1}^{l} K^c_{zj} Y_{jc} + \sum_{j=l+1}^{n} K^c_{zj} f^*_{jc}}{\sum_{j=1}^{n} |K^c_{zj}|}. \tag{15}$$

Note that the diagonal entries K^c_{ii} $(1 \leqslant i \leqslant n)$ do not influence the solution $\boldsymbol{f}^*_{v,c}$ in Eq. (11) because they are counteracted in calculating the matrix B^c. Moreover, K^c_{zz} is not involved in the solution f^*_{zc} in Eq. (15). Consequently, we can ignore the diagonal elements of the learned class-specific affinity (or kernel) matrix K^c.

3.5 Algorithm

In the sequel, we give the whole algorithmic framework for multi-class semi-supervised classification in Algorithm 1. We still call the algorithm *Bidirectional Label Propagation* (BLP). Since BLP is able to cope with both training and testing data, it is truly semi-supervised, *i.e.*, not only transductive but also inductive. For binary-class problems ($C = 2$), only once graph learning together with once label propagation is required.

4 Experiments

In this section, we evaluate the proposed novel graph-based SSL algorithm bidirectional label propagation (BLP), which integrates class-specific graph learning and multi-class label propagation, on one toy problem and two real-world datasets. We compare BLP with the state-of-the-art graph-based SSL algorithms including *Local and Global Consistency* (LGC)[24], *Quadratic Criterion* (QC)[3], *Gaussian Fields and Harmonic Functions* (GFHF) plus the postprocessing operation *Class Mass Normalization* (CMN)[26], *Laplacian Regularized Least Squares* (LapRLS)[2],

and *Laplacian Support Vector Machines* (LapSVMs)[2], all of which can directly be applied to multi-class problems.

To entail a fair comparison, we use Eq. (1) (adopting an empirical choice of σ suggested in Subsection 3.1) to build the same k-NN graph for all algorithms on each dataset. The width of the RBF kernel for LapRLS and LapSVM is set by cross validation. In practice, GFHF often exhibits more robust performance than LGC because of the hard labeling constraint, and CMN usually further improves the performance of GFHF. The regularization parameters associated with LGC, QC, LapRLS and LapSVM are tuned to the best. As an advantage, our algorithm BLP is less sensitive to the two parameters ε and ξ. Throughout our experiments, we fix them to 10^{-3} and 10^{-6}, respectively.

4.1 Toy problem

We first conduct experiments on one synthetic dataset plotted in Fig. 3. This dataset is Noisy Face Contour, which is composed of 266 points belonging to three classes and 61 uniformly distributed noisy points. We do not care about the labels of the noisy points, so we compute classification error rates only on non-noise points whose labels are known in advance.

The existing graph-based SSL algorithms result in worse classification results as the noisy points essentially destroy the graph structure so that labels are unnecessarily propagated along them. Shown in Fig. 3, our algorithm BLP exhibits a fully correct classification when only one point of each class is labeled, whereas all of the other competing algorithms give mistakes. We do not show the visual classification results achieved by other algorithms due to the space limit. We further show average error rates over 100 random trials in Table 1. We test all compared algorithms with three and six initially labeled points, respectively. BLP clearly demonstrates a substantial advantage over all of the other algorithms whether using a 5-NN graph or a 10-NN graph. Therefore, we can say that the proposed graph learning mechanism and the BLP criterion are robust to noise.

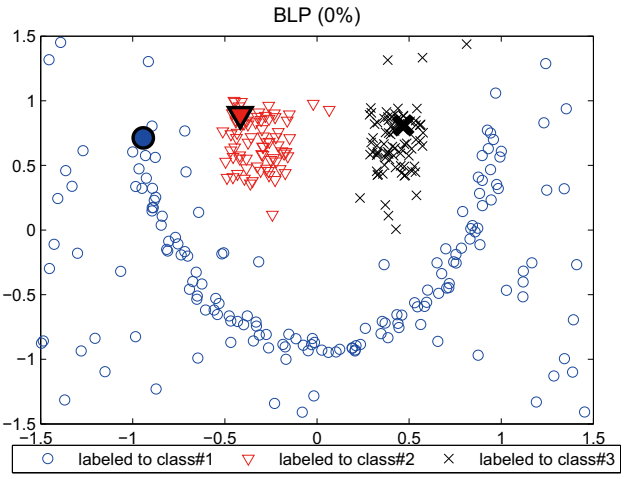

Figure 3. The semi-supervised classification result achieved by our proposed SSL algorithm BLP on the toy problem: Noisy Face Contour.

Table 1 Average classification error rates on the toy problem

Error Rate	Toy: 3 labeled		Toy: 6 labeled	
(%)	5-NN Graph	10-NN Graph	5-NN Graph	10-NN Graph
LGC	9.28±4.64	8.55±3.80	5.54±4.84	5.70±4.23
QC	10.78±4.41	9.61±3.29	6.27±5.14	6.37±4.07
GFHF	7.14±5.75	8.61±5.15	4.18±4.53	4.67±4.44
GFHF+CMN	7.14±3.75	7.94±2.67	5.04±4.00	5.51±3.74
LapRLS	6.77±5.11	8.79±5.02	4.21±4.45	5.10±4.48
LapSVM	6.65±5.39	8.40±4.81	4.22±4.44	5.08±4.43
BLP	**2.22±2.41**	**3.81±3.06**	**0.19±1.06**	**1.86±2.57**

4.2 Handwritten digit recognition

We evaluate these graph-based SSL algorithms on the `USPS` (test subset) handwritten digits dataset in which each example is a 16×16 image and there are ten types of digits "0, 1, 2, ..., 9" used as ten classes. There are 160 digit images in each class at least, summing up to a total of 2007 examples. Fig. 5 shows ten examples. We randomly choose initially labeled examples such that they contain at least one labeled example from each class. Averaged over 20 trials, we calculate the error rates for all referred algorithms with the number of the initially labeled examples increasing from 20 to 100. The classification results are displayed in Fig. 4 and Table 2. Again, we observe that BLP is significantly superior to the other compared algorithms, which demonstrates that the class-specific graph learning scheme and the more sophisticated bidirectional label propagation technique, which exploits both similarities and dissimilarities hidden among the input examples, boost graph-based SSL prominently.

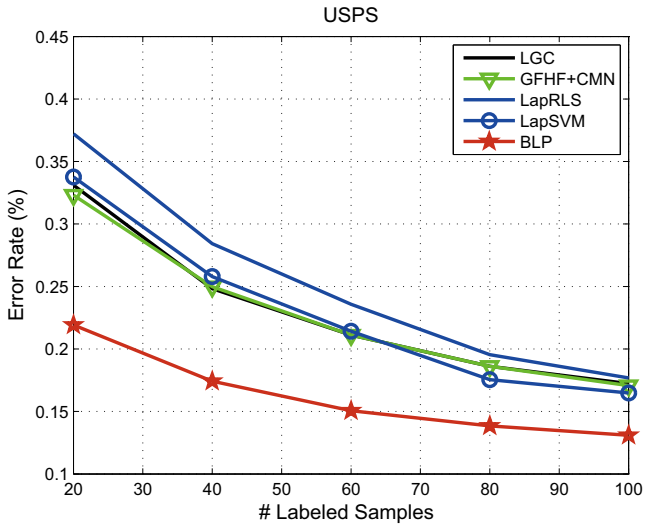

Figure 4. Average error rates vs. numbers of initially labeled examples.

Table 2 Average classification error rates on the USPS dataset

Error Rate	USPS: 20 labeled		USPS: 100 labeled	
(%)	10-NN Graph	20-NN Graph	10-NN Graph	20-NN Graph
LGC	36.40±5.40	33.10±5.40	19.50±1.43	17.23±1.29
QC	37.15±5.43	34.24±5.18	21.35±1.33	18.77±1.22
GFHF	60.65±7.80	57.28±7.81	25.86±3.10	22.04±2.64
GFHF+CMN	34.98±5.33	32.31±5.41	19.25±2.15	17.07±1.89
LapRLS	37.21±5.20	37.22±5.28	18.98±1.89	17.68±1.86
LapSVM	34.92±4.94	33.76±5.24	18.16±1.80	16.48±1.85
BLP	**24.65±4.85**	**21.92±4.66**	**14.54±1.00**	**13.10±1.30**

4.3 Face recognition

Now we draw our attention to the intensively studied topic, face recognition. Our experiments are performed on a subset of 3160 facial images selected from 316 persons in the FRGC version 2[17]. We align all these faces according to the positions of eyes and mouth and then crop them to the fixed size of 64×72 pixels. We adopt grayscale values of these images as facial features. Fig. 5 displays ten face image examples.

Figure 5. Examples: ten face images in the top line come from one person of the FRGC dataset, and ten digit images in the second line come from the USPS dataset.

We randomly choose 316 $\sim$ 1000 images in this dataset as the initially labeled examples, and keep the remaining examples as the unlabeled data. The chosen labeled examples always cover the total 316 classes (*i.e.*, persons).

By repeating the recognition process 20 times, we plot average recognition rates of five compared SSL algorithms using the same 6-NN graph according to the expanding initially labeled data size in Fig. 6. The results in Fig. 6 further confirm the effectiveness of the proposed BLP algorithm which outperforms all of the other compared SSL algorithms.

We show that for this high-dimensional multi-class classification task, both of the proposed graph learning procedure (*i.e.*, class-specific affinity matrix learning) and the ingenious label propagation technique that works along both positive and negative edge directions exhibit substantial robustness on this challenging multi-class problem, whereas the other SSL algorithms either suffer from non-trivial high-dimensional noise or are vulnerable to multiple classes.

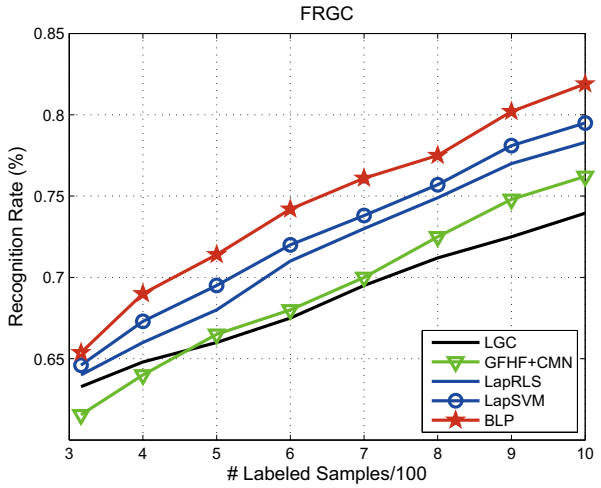

Figure 6. Average recognition rates vs. numbers of initially labeled examples.

5 Conclusion and Discussion

Graph-based methods form a main category of state-of-the-art semi-supervised learning, offering flexibility and easy implementation for broad applications. To this end, we follow previous methods and present a novel graph-theoretical semi-supervised learning framework. In this framework, we first incorporate initial sparse label information to learn discriminative graphs that are specific to each class and reveal the dissimilarities inherent among the given examples. By simultaneously leveraging similarities and dissimilarities, we then develop a novel label propagation technique to propagate labels along positive and negative edges that the learned graphs produce. Our proposed bidirectional label propagation method can ensure consistent labelings along the positive edges and inconsistent labelings along the negative edges. We have shown through the experiments on synthetic and real-world datasets that our approach is robust against noise present classification problems and effective for multi-class classification tasks.

In our future research plan, we intend to study learning discriminative graphs for wider applications such as learning semantic-aware graphs for web image reranking[14] and learning large-scale graphs for web-scale data collections[15]. On the theory side, we also pursue to investigate the potential connections between graph learning and hash code generation as advocated by Ref. [16], or manifold methods such as manifold ranking[22] and manifold discovery[23].

References

[1] Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation, 2003, 15(6): 1373–1396.

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

[3] Bengio Y, Dellalleau O, Roux NL. Label propagation and quadratic criterion. In: Chapelle O, Schölkopf B, Zien A eds. Semi-Supervised Learning. The MIT Press. 2006.

[4] Chapelle O, Sindhwani V, Keerthi SS. Optimization techniques for semi-supervised support vector machines. Journal of Machine Learning Research, 2008, 9: 203–233.

[5] Chung F. Spectral graph theory. American Mathematical Society: CBMS Regional Conference Series in Mathematics. 1997.
[6] Collobert R, Sinz FH, Weston J, Bottou L. Large scale transductive SVMs. Journal of Machine Learning Research, 2006, 7: 1687–1712.
[7] Delalleau O, Bengio Y, Roux NL. Efficient non-parametric function induction in semi-supervised learning. Proc. of AI and Statistics. 2005.
[8] Goldberg AB, Zhu X, Wright SJ. Dissimilarity in graph-based semi-supervised classification. Proc. of AI and Statistics. 2007.
[9] Hein M, Maier M. Manifold denoising. Advances in Neural Information Processing Systems, 2006, 19.
[10] Joachims T. Transductive inference for text classification using support vector machines. Proc. of International Conference on Machine Learning. 1999.
[11] Karlen M, Weston J, Erkan A, Collobert R. Large scale manifold transduction. Proc. of International Conference on Machine Learning. 2008.
[12] Liu W, Chang SF. Robust multi-class transductive learning with graphs. Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2009.
[13] Liu W, He J, Chang SF. Large graph construction for scalable semi-supervised learning. Proc. of International Conference on Machine Learning. 2010.
[14] Liu W, Jiang YG, Luo J, Chang SF. Noise resistant graph ranking for improved web image search. Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2011.
[15] Liu W, Wang J, Chang SF. Robust and scalable graph-based semi-supervised learning. Proc. of the IEEE, 2012, 100(9): 2624–2638.
[16] Liu W, Wang J, Kumar S, Chang SF. Hashing with graphs. Proc. of International Conference on Machine Learning. 2011.
[17] Phillips PJ, Flynn PJ, Scruggs T, Bowyer KW, Chang J, Hoffman K, Marques J, Min J, Worek W. Overview of the Face Recognition Grand Challenge. Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2005.
[18] Rasmussen CE, Williams CKI. Gaussian Processes for Machine Learning. The MIT Press, 2006.
[19] Sindhwani V, Niyogi P, Belkin M. Beyond the point cloud: from transductive to semi-supervised learning. Proc. of International Conference on Machine Learning. 2005.
[20] Tong W, Jin R. Semi-supervised learning by mixed label propagation. Proc. of AAAI Conference on Artificial Intelligence. 2007.
[21] Wang J, Wang F, Zhang C, Shen HC, Quan L. Linear neighborhood propagation and its applications. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2009, 31(9): 1600–1615.
[22] Xu B, Bu J, Chen C, Cai D, He X, Liu W, Luo J. Efficient manifold ranking for image retrieval. Proc. of International ACM SIGIR Conference on Research and Development in Information Retrieval. 2011.
[23] Zhang T, Ji R, Liu W, Tao D, Hua G. Semi-supervised learning with manifold fitted graphs. Proc. of International Joint Conference on Artificial Intelligence. 2013.
[24] Zhou D, Bousquet O, Lal T, Weston J, Schölkopf B. Learning with local and global consistency. Advances in Neural Information Processing Systems, 2003, 16.
[25] Zhu X. Semi-supervied learning literature survey. Computer Science Technical Report. University of Wisconsin, Madison. 2008.
[26] Zhu X, Ghahramani Z, Lafferty J. Semi-supervised learning using gaussian fields and harmonic functions. Proc. of International Conference on Machine Learning. 2003.