Large Margin Graph Construction for Semi-Supervised Learning

Lan-Zhe Guo[†] and Shao-Bo Wang[†] and Yu-Feng Li[*]
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China
{guolz, wangsb, liyf}@lamda.nju.edu.cn

***Abstract*—Graph-based semi-supervised learning (GSSL) has gained increased interests in the last few years. A large number of empirical results show that the performance of GSSL methods heavily depends on the graph construction approach. Although great efforts have been devoted to construct good graphs, it remains challenging to construct a good graph in general situations. To alleviate this problem, this paper presents a novel graph construction approach. Unlike previous approaches that typically optimize a kNN-type loss on the unlabeled data, the proposed approach further enforces that the prediction of unlabeled data has a large margin separation so as to help exclude low-quality graphs. We formulate the problem as an optimization and present an efficient algorithm. Experimental results on benchmark data sets show that the proposed approach has a stronger ability to construct good graphs comparing with several representative graph construction approaches.**

I. Introduction

With the data explosion in recent years, unlabeled data has increased much more rapidly than labeled data which requires extra human labor. With the ability to utilize unlabeled data to improve performance, semi-supervised learning (SSL) has aroused more and more interests. Among different kinds of SSL paradigms, graph-based semi-supervised learning (GSSL) attracts significant attention since proposed and has been widely applied in a large number of applications [1] because of its advantages such as closed-form solution, easy implementation and promising performance.

The key to GSSL methods is its basic assumption: data lies in an underlying manifold and closer points are more likely share the same class. To approximate the underlying manifold, a graph is constructed where a node corresponds to an instance and a pair of nodes are connected by an edge. After a graph is constructed, GSSL methods perform a process called label inference on the graph where only a few nodes are associated with labels to finally predict the labels of unlabeled data. It's obvious that the prediction performance heavily depends on the graph construction approach. Using a low-quality graph can degenerate performance and leads to unsafe problem [2], [3]. The graph construction problem has been a consensus in the research community recently [4], [5], [6]. Moreover, a high-quality graph can also benefit other tasks, such as feature selection [7] or matrix completion [8].

Generally speaking, graph construction involves two important choices. First, the user chooses a similarity function or kernel for estimating the affinity between all pairs of instances, such as Gaussian kernel [9]. Second, the user chooses a sparsification method to obtain a sparse weighted subgraph from the fully connected weighted graph. Sparsity is necessary to GSSL for being efficient and robust [1]. Unlike the choice of the similarity function, it is generally hard to say whether a specific sparsification method can get a good graph. The most popular choice is the kNN method because of its concision and effectiveness. In a kNN graph, each node has edges to its first k nearest neighbors. εNN graph is another choice where there is an edge between a pair of nodes if their distance is less than ε, while it may be unstable in some cases. As a most representative graph construction approach, b-matching [5] guarantees a regular graph where all nodes have the same degree b in contrast to the irregularity of kNN graph. With experimental results, the authors argue that a regular graph can achieve better classification results compared to kNN. However, building a b-matching graph is usually impractical in terms of computational cost. As another way of graph construction, Argyriou *et al.* [10] propose a graph construction approach which combines multiple graphs constructed with a variety of distances functions and the 'k' in nearest neighbors. Despite its superior performance in some cases, this method has heavy computational cost and unstable performance. Despite the efforts devoted by existing research works to construct good graphs, it's still an open problem that how to construct a good graph in general situations.

Recently, Li *et al.* [11] proposed a large margin criterion to judge the quality of the graph. They performed a series of experiments to show the effectiveness of the large margin assumption on predictive values of good graphs. More specifically, when a graph owns a high quality, its prediction on unlabeled data may have a large margin separation. Following their conclusion, in this paper, we propose a graph construction approach by considering the large margin assumption. Unlike previous approaches that typically optimize a kNN-type loss on the unlabeled data, the proposed approach also optimizes a margin-type loss so as to learn a graph that has large margin separation on its prediction. We formulate the learning problem as an optimization problem and present an efficient algorithm to solve the problem. To the best of our knowledge, there have no existing graph construction approaches utilizing

[†] Contributed equally
[*] Corresponding author

the large margin assumption before.

The rest of this paper is organized as follows. Section II presents the proposed method. Section III gives an efficient algorithm to solve the optimization problem. Empirical results are reported in Section IV, finally, we conclude this paper in Section VI.

II. The Proposed Method

In GSSL, given a few of labeled instances $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and a large amount of unlabeled instances $\{\mathbf{x}_{l+i}\}_{i=1}^{u}$ (typically $l \ll u$) where $y \in \{\pm 1\}$ is the output label for the input instance $\mathbf{x} \in \mathbb{R}^d$. Let $G = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ denotes a graph. Here $\mathcal{V}$ is a set of $n = l + u$ nodes in which each node corresponds to an instance. $\mathcal{E}$ is a set of undirected edges between pairs of nodes. $\mathbf{W} \in \mathbb{R}^{n \times n}$ is a nonnegative and symmetric adjacency matrix associating with $\mathcal{V}$ and $\mathcal{E}$ where each element W_{ij} is the weight of the edge $e_{ij} \in \mathcal{E}$ and reflects the affinity between $\mathbf{x}_i$ and $\mathbf{x}_j$. The sparsity of a graph means the number of edges $|\mathcal{E}| \ll \frac{n(n-1)}{2}$ *i.e.*, $\mathbf{W}$ is a sparse matrix.

Given a graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, GSSL aims to infer the labels $\{y_{l+i}\}_{i=1}^{u}$ of unlabeled instances $\{\mathbf{x}_{l+i}\}_{i=1}^{u}$ by minimizing a kNN-type loss as follows (ignoring specific constraints):

$$\min_{\mathbf{f}\in\mathbb{R}^n} \frac{1}{2}\sum_{i,j=1}^{n} W_{ij}(f_i - f_j)^2 \tag{1}$$

where f_i is the predictive value for $\mathbf{x}_i$. Specific label inference methods may have different constraints to control the value of $\mathbf{f}$, such as f_i for labeled data is exactly equal to y_i or f_i and y_i need to be close for labeled data but not necessarily be the same. Finally, GSSL predicts the labels by outputting $y_i = \text{sign}(f_i)$ for unlabeled data.

In [11], authors proposed a large margin assumption with respect to the predictive value of GSSL. Specifically, suppose we obtain two sets of prediction $\mathbf{f}^{(1)}$ and $\mathbf{f}^{(2)}$ from two different graphs $G^{(1)}$ and $G^{(2)}$ respectively by performing one GSSL algorithm, the assumption indicates that if $\mathbf{f}^{(1)}$ has a larger margin separation than $\mathbf{f}^{(2)}$, then $G^{(1)}$ may be a better graph than $G^{(2)}$ with high probability and vice versa.

Based on the large margin assumption, we present a graph construction approach to take into account the margin-type loss in order to learn a good graph whose prediction has a large margin separation. The idea is that a good graph can be constructed by minimizing a margin-type loss of its prediction. All the proposed method do is like to select the graph with the largest margin on its prediction from multiple candidate graphs, only that the candidates are from a predefined domain.

Typically, hinge loss is used to reflect how much the model violates the large margin assumption. However, for the concision of formulation and convenience of optimization, we adopt least square loss instead of hinge loss. It has been theoretically proved that least square loss corresponds to the hard margin while hinge loss corresponds to the soft margin [12].

In order to learn a good graph, we formulate the learning problem as an optimization problem:

$$\min_{\mathbf{W}\in\mathcal{W}} \min_{\mathbf{f}\in\mathcal{R},\hat{\mathbf{y}}\in\mathcal{B}} \frac{\frac{1}{2}\sum_{i,j=1}^{n} W_{ij}(f_i - f_j)^2}{\frac{1}{2}\sum_{i,j=1}^{n} W_{ij}} + \frac{\lambda}{u}\sum_{i=l+1}^{n}(f_i - \hat{y}_i)^2 \tag{2}$$

where $|\mathcal{E}| = \frac{1}{2}\sum_{i,j=1}^{n}\mathbb{I}(W_{ij} > 0)$ and u are normalizer, λ is a parameter to trade off two different kinds of losses, and $\hat{\mathbf{y}}$ is a set of pseudo labels. Let $\mathbb{S}$ denotes the set of symmetric matrics, then $\mathcal{W} = \{\mathbf{W} \in \mathbb{S}^{n\times n} | \text{diag}(\mathbf{W}) = \mathbf{0}; (\mathbf{W} - \mathbf{A}) \circ \mathbf{W} = \mathbf{0}; \frac{1}{2}\sum_{i,j=1}^{n}\mathbb{I}(W_{ij} > 0) = C; \text{ncc}(\mathbf{W}) = 1\}$, where $\circ$ denotes the Hadamard product and $C \geq n - 1$ is a super parameter to control the sparsity of the graph, ncc($\mathbf{W}$) denotes the number of connected components. With $L = \{1, 2, \ldots, l\}$ and $U = \{l+1, l+2, \ldots, l+u\}$, $\mathcal{R} = \{\mathbf{f} \in \mathbb{R}^n | \mathbf{f}_L = \mathbf{y}_L\}$, $\mathcal{B} = \{\hat{\mathbf{y}} \in \{\pm 1\}^n | \hat{\mathbf{y}}_L = \mathbf{y}_L; \frac{1}{u}\sum_{i\in U}\hat{y}_i = \frac{1}{l}\sum_{i\in L}\hat{y}_i\}$, where $\frac{1}{u}\sum_{i\in U}\hat{y}_i = \frac{1}{l}\sum_{i\in L}\hat{y}_i$ constrains the ratio of classes in order to avoid ill solver.

For $\mathcal{W}$, $(\mathbf{W}-\mathbf{A})\circ\mathbf{W} = \mathbf{0}$ constrains the exact value of each non-zero element in $\mathbf{W}$, *i.e.*, $W_{ij} = A_{ij}$ or $W_{ij} = 0$, where $\mathbf{A}$ is an affinity matrix which is given by users. There are several approaches to estimate the affinity between pairs of nodes. The simplest approach is the binary weighting approach, *i.e.*, $A_{ij} = 0/1$. In this case, $\mathbf{W}$ is actually associated with an unweighted graph. An alternative approach is Gaussian kernel, which is defined as $A_{ij} = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$. Moreover, the cosine distance is also commonly used.

Let $\mathbf{W} = \mathbf{P}\circ\mathbf{A}$ where $\mathbf{P}$ is an indicator matrix with binary-valued elements, then Eq.(2) can be rewrite as

$$\min_{\mathbf{P}\in\mathcal{P}} \min_{\mathbf{f}\in\mathcal{R},\hat{\mathbf{y}}\in\mathcal{B}} \frac{\sum_{i,j=1}^{n} P_{ij}A_{ij}(f_i - f_j)^2}{\sum_{i,j=1}^{n} P_{ij}} + \frac{\lambda}{u}\sum_{i=l+1}^{n}(f_i - \hat{y}_i)^2 \tag{3}$$

where $\mathcal{P} = \{\mathbf{P} \in \{0,1\}^{n\times n} | \mathbf{P} = \mathbf{P}^{\mathrm{T}}; \text{diag}(P) = \mathbf{0}; \sum_{i,j=1}^{n} P_{ij} = 2C; \text{ncc}(\mathbf{P}) = 1\}$.

III. Algorithm

To solve the objective in Eq.(2), we use an alternative optimization method. It alternatively optimizes the variable $\mathbf{W}$ (or $\{\mathbf{f}, \hat{\mathbf{y}}\}$) when fixing $\{\mathbf{f}, \hat{\mathbf{y}}\}$ (or $\mathbf{W}$) as constants. Specifically, when $\mathbf{W}$ is fixed, we further employ an alternating optimization *w.r.t.* $\mathbf{f}$ and $\hat{\mathbf{y}}$. Since the objective in Eq.(2) is jointly convex for $\{\mathbf{f}, \hat{\mathbf{y}}\}$ when $\mathbf{W}$ is fixed, the subproblems of alternating optimization are convex for both $\mathbf{f}$ and $\hat{\mathbf{y}}$.

More specifically, when $\mathbf{f}$ is fixed, according to [13], it is known that the rank of the elements in $\hat{\mathbf{y}}$ is consistent to that of the elements in $\mathbf{f}$. Moreover, since we constrain that $\hat{\mathbf{y}}_L = \mathbf{y}_L$, the optimal solution of $\hat{\mathbf{y}}_U$ could be solved in a closed-form solution:

$$y_{l+j} = \begin{cases} +1 & r_j \leq \frac{u}{l}\sum_{i=1}^{l}\mathbb{I}(y_i = +1) \\ -1 & \text{otherwise} \end{cases} \tag{4}$$

where $\{r_1, \ldots, r_u\}$ are the ranks of the predictions on the unlabeled instances $\{\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}\}$ (sorted in a descending order).

When $\hat{\mathbf{y}}$ is fixed, the inner minimization problem of Eq.(2) is equivalent to the following form:

$$\min_{\mathbf{f}\in\mathcal{R}} \frac{\mathbf{f}^{\mathrm{T}}\mathbf{L}\mathbf{f}}{C} + \frac{\lambda}{u}\|\mathbf{f}_U - \hat{\mathbf{y}}_U\|^2 \tag{5}$$

where $\mathbf{L}$ is the graph Laplacian defined as $\mathbf{L} = \mathrm{diag}(\mathbf{W}\mathbf{1}) - \mathbf{W}$.

Since $\mathbf{f} = [\mathbf{f}_L^{\mathrm{T}}, \mathbf{f}_U^{\mathrm{T}}]^{\mathrm{T}}$ and we constrain $\mathbf{f}_L = \mathbf{y}_L$, the Laplacian $\mathbf{L}$ can be partitioned into four blocks:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{LL} & \mathbf{L}_{LU} \\ \mathbf{L}_{UL} & \mathbf{L}_{UU} \end{bmatrix} \tag{6}$$

Eq.(5) can be rewritten as

$$\min_{\mathbf{f}_U\in\mathbb{R}^u} \frac{2}{C}\mathbf{y}_L^{\mathrm{T}}\mathbf{L}_{LU}\mathbf{f}_U + \frac{1}{C}\mathbf{f}_U^{\mathrm{T}}\mathbf{L}_{UU}\mathbf{f}_U + \frac{\lambda}{u}\mathbf{f}_U^{\mathrm{T}}\mathbf{f}_U - \frac{2\lambda}{u}\hat{\mathbf{y}}_U^{\mathrm{T}}\mathbf{f}_U \tag{7}$$

Let $G(\mathbf{f}_U)$ denote the objective in Eq.(7), then

$$\nabla G = \frac{2}{C}\mathbf{L}_{UL}\mathbf{y}_L + \frac{2}{C}\mathbf{L}_{UU}\mathbf{f}_U + \frac{2}{u}\lambda\mathbf{f}_U - \frac{2}{u}\lambda\hat{\mathbf{y}}_U \tag{8}$$

Since $\nabla^2 G \succeq 0$, by setting ∇G to 0 we can obtain the global solution

$$\mathbf{f}_U = (\frac{1}{C}\mathbf{L}_{UU} + \frac{\lambda}{u}\mathbf{I})^{-1}(\frac{\lambda}{u}\hat{\mathbf{y}}_U - \frac{1}{C}\mathbf{L}_{UL}\mathbf{y}_L) \tag{9}$$

When $\{\mathbf{f}, \hat{\mathbf{y}}\}$ are fixed, the optimization problem of Eq.(3) is equivalent to the following form:

$$\min_{\mathbf{P}\in\mathcal{P}} \sum_{i,j=1}^{n} P_{ij}A_{ij}(f_i - f_j)^2 \tag{10}$$

which can be effectively solved by Minimum Spanning Tree algorithm. Specifically, a symmetric matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ is constructed according to $\mathbf{A}$ and $\mathbf{f}$ such that $M_{ij} = A_{ij}(f_i - f_j)^2$. Then a minimum spanning tree $G' = \{\mathcal{V}', \mathcal{E}', \mathbf{W}'\}$ is constructed by taking $\mathbf{M}$ as the input. Set M_{ij} and M_{ji} to 0 for each $e_{ij} \in \mathcal{E}'$ (note e_{ij} is undirected while $\mathbf{M}$ is symmetric). Then every time select the minimal non-zero element M_{ij} in $\mathbf{M}$, set M_{ij} and M_{ji} to 0, and add corresponding edge e_{ij} to $\mathcal{E}'$ until $|\mathcal{E}'|$ increases to C. Finally, let $\mathbf{P}$ be the unweighted adjacency matrix of $\mathcal{E}'$ and it is the solution of Eq.(10).

After alternating optimization converges, an expected large margin graph is constructed as $\mathbf{W} = \mathbf{P} \circ \mathbf{A}$. The pseudo code of the whole proposed method is given in Algorithm 1.

IV. EXPERIMENT

A. Experimental Setting

We perform the experiments on 6 UCI data sets[1]. We do experiment on noisy graphs with noisy ratio range from $[0, 0.5]$. For the GSSL method, we adopt the classical method: the Class Mass Normalization (CMN) [9]. For the graph construction method, we compared with noisy kNN, randomly denoise and LMG with $\lambda = 0$ and $\lambda = 0.1$. For each data set, we random select $l = 10$ labeled instances and the others are treated as unlabeled instances. Experimental results are obtained from 30 repeated runs for each case.

Algorithm 1 Large Margin Graph Construction

```
Input: labeled instances $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, unlabeled instances $\{\mathbf{x}_i\}_{i=l+1}^{l+u}$, affinity matrix $\mathbf{A}$, initial sparsification matrix $\mathbf{P}$ (generally a $k$NN graph is recommended), model parameters $C$ and $\lambda$.
Output: an adjacency matrix $\mathbf{W}$ associated with a weighted undirected graph.
Initialize $\mathbf{W} = \mathbf{P} \circ \mathbf{A}$ and $\mathbf{f}_L = \mathbf{y}_L$, where $L = \{1, \dots, l\}$
Perform a GSSL algorithm on the adjacency matrix $\mathbf{W}$ to obtain a set of pseudo labels $\hat{\mathbf{y}}_U$ for the unlabeled instances, where $U = \{l+1, \dots, l+u\}$
repeat
    repeat
        Fix $\hat{\mathbf{y}}_U$ and update the solution of $\mathbf{f}_U$ via Eq.(9)
        Fix $\mathbf{f}_U$ and update the solution of $\hat{\mathbf{y}}_U$ via Eq.(4)
    until the objective of Eq.(2) does not decrease
    Construct $\mathbf{M}$ such that $M_{ij} = A_{ij}(f_i - f_j)^2$
    Perform the Minimum Spanning Tree algorithm on $\mathbf{M}$ to obtain an unweighted symmetric adjacency matrix $\mathbf{P}$
    If $P_{ij} = 1$ then $M_{ij} = 0$ for $i, j = 1, \dots, l+u$
    while $\frac{1}{2}\sum_{i,j=1}^{l+u} P_{ij} < C$ do
        Select minimal non-zero element $M_{ij}$ of $\mathbf{M}$
        Set $P_{ij} = P_{ji} = 1$ and $M_{ij} = M_{ij} = 0$
    end while
    Set $\mathbf{W} = \mathbf{P} \circ \mathbf{A}$
until the objective of Eq.(2) does not decrease
return $\mathbf{W}$
```

B. Performance with Noisy Graphs

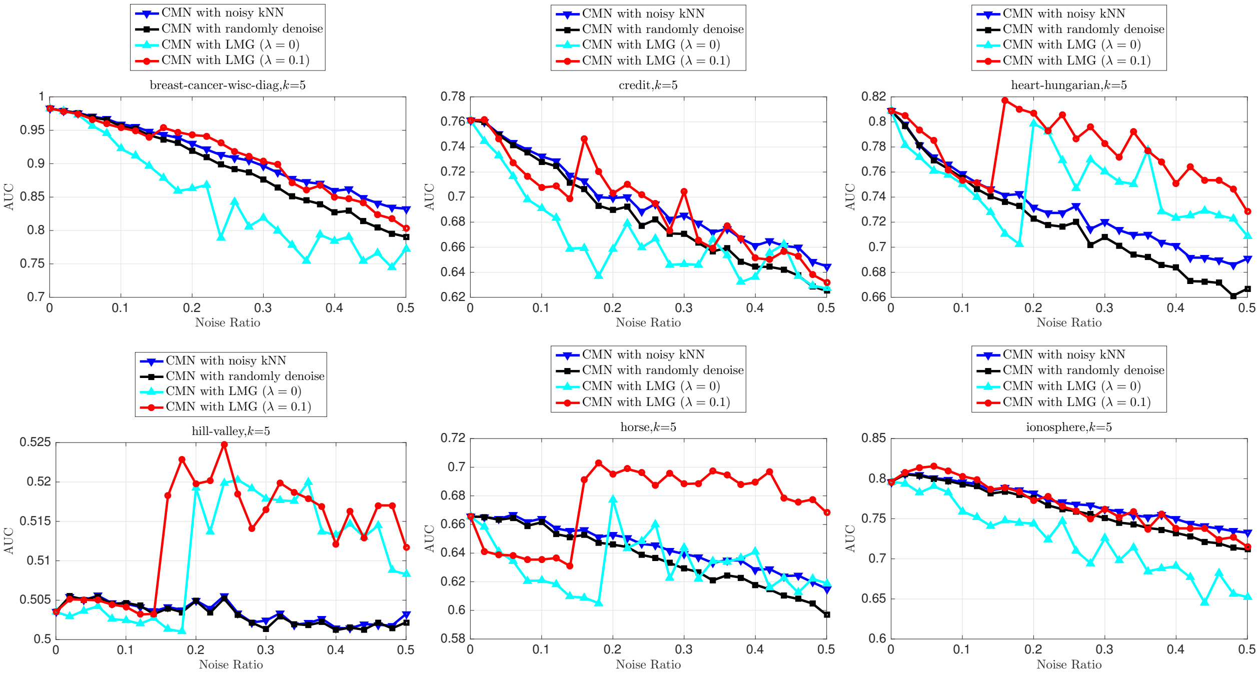
Experimental results with noisy graphs are shown in Figure 1. From the experiment results, we can see that the proposed method is always better than randomly denoise method and comparable with noisy kNN. In three cases, the proposed method achieves a significant performance gain. Moreover, the red line (CMN with LMG ($\lambda = 0.1$)) is always better than the cyan line (CMN with LMG ($\lambda = 0$)). These validate the effectiveness proposed method. Besides, there is another thing need to be mentioned: randomly removing edges may lead to a graph that has more than one connected components, while the proposal does not suffer this problem.

V. LARGE MARGIN GRAPH CONSTRUCTION V.S. GRAPH PRESENTATION LEARNING

Graph representation learning, i.e., learning a low dimensional vector representation of nodes in graphs, has attracted significant attention in recent years and plays a critical role in the area of GSSL. Meantime, large margin principle is also widely used to train a robust classifier and help avoid overfitting issues, which is particularly useful when labeled data is limited. However, to our best knowledge, the large-margin principle has rarely been applied to graph-based methods. In this paper, we proposed a graph construction method based on large margin principle and show its effectiveness with empirical results. It demonstrates that the large margin principle can work well with graph-based methods. It is innovative for the

[1] http://archive.ics.uci.edu/ml/datasets.html

Fig. 1. Performance of the proposed method with noisy graphs. '$\lambda = 0$' refers to that optimization objective only consists of kNN-type loss. 'Randomly denoise' refers to that randomly remove edges of the noisy graph until the number of edges down to $C = \frac{1}{2}\sum_{i,j=1}^{n} W_{ij}^{knn}$.



algorithm design of graph representation learning. We think that one key reason for why large margin principle works well with graph-based methods is that, the underlying assumption for large margin principle (large-margin assumption) and graph-based methods (manifold assumption) are kind of complementary. Specifically, manifold assumption emphasizes that the data closeness within same classes, whereas ignores the data separability between different classes. By contrast, large margin assumption emphasizes the data separability, but ignores the data closeness. Therefore, by taking the two assumptions into account simultaneously, one can encourage the inter-class separability between learned features for graphs and leads to a better decision boundary.

VI. Conclusion

In this paper, we study the graph construction problem in GSSL, a key component for GSSL performance, and develop a novel graph construction method by considering the margin assumption. The proposed methods optimize a margin-type loss to learn a graph that has a large margin separation on its prediction and can be formulated as an alternative optimization problem which can be solved efficiently. Extensive experimental results validate the effectiveness of the proposed method.

Acknowledgment

The research was supported by the National Key R&D Program of China (2017YFB1001903), the National Natural Science Foundation of China (61772262) and the Fundamental Research Funds for the Central Universities (020214380044).

References

[1] W. Liu, J. Wang, and S.-F. Chang, "Robust and scalable graph-based semisupervised learning," *Proceedings of the IEEE*, vol. 100, pp. 2624–2638, 2012.

[2] Y.-F. Li, J. T. Kwok, and Z.-H. Zhou, "Towards safe semi-supervised learning for multivariate performance measures." in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2016, pp. 1816–1822.

[3] L.-Z. Guo and Y.-F. Li, "A general formulation for safely exploiting weakly supervised data," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018.

[4] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, pp. 55–67, 2008.

[5] T. Jebara, J. Wang, and S.-F. Chang, "Graph construction and b-matching for semi-supervised learning," in *Proceedings of the 26th International Conference on Machine Learning*, 2009, pp. 441–448.

[6] D.-M. Liang and Y.-F. Li, "Learning safe graph construction from multiple graphs," in *International CCF Conference on Artificial Intelligence*, 2018, pp. 41–54.

[7] Z. Zhao, L. Zhang, X. He, and W. Ng, "Expert finding for question answering via graph regularized matrix completion," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, pp. 993–1004, 2015.

[8] Z. Zhao, X. He, D. Cai, L. Zhang, W. Ng, and Y. Zhuang, "Graph regularized feature selection with data reconstruction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, pp. 689–700, 2016.

[9] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International Conference on Machine Learning*, 2003, pp. 912–919.

[10] A. Argyriou, M. Herbster, and M. Pontil, "Combining graph laplacians for semi–supervised learning," in *NIPS*, 2006, pp. 67–74.

[11] Y.-F. Li, S.-B. Wang, and Z.-H. Zhou, "Graph quality judgement: A large margin expedition," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2016, pp. 1725–1731.

[12] J. Ye and T. Xiong, "Svm versus least squares svm," in *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, 2007.

[13] Y.-F. Li, I. W. Tsang, J. T. Kwok, and Z.-H. Zhou, "Convex and scalable weakly labeled svms," *Journal of Machine Learning Research*, pp. 2151–2188, 2013.