Front. Comput. Sci., 2018, 12(4): 725–735
https://doi.org/10.1007/s11704-017-6543-5

RESEARCH ARTICLE

Instance selection method for improving graph-based semi-supervised learning

Hai WANG[1,2], **Shao-Bo WANG**[1,2], **Yu-Feng LI** (✉)[1,2]

1 National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
2 Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210023, China

Abstract Graph-based semi-supervised learning is an important semi-supervised learning paradigm. Although graph-based semi-supervised learning methods have been shown to be helpful in various situations, they may adversely affect performance when using unlabeled data. In this paper, we propose a new graph-based semi-supervised learning method based on instance selection in order to reduce the chances of performance degeneration. Our basic idea is that given a set of unlabeled instances, it is not the best approach to exploit all the unlabeled instances; instead, we should exploit the unlabeled instances that are highly likely to help improve the performance, while not taking into account the ones with high risk. We develop both transductive and inductive variants of our method. Experiments on a broad range of data sets show that the chances of performance degeneration of our proposed method are much smaller than those of many state-of-the-art graph-based semi-supervised learning methods.

Keywords graph-based semi-supervised learning, performance degeneration, instance selection

1 Introduction

In many applications, there are abundant unlabeled training data whereas the acquisition of class labels is costly and difficult. For example, in webpage categorization [1], manually labeled webpages constitute a very small part of the entire web, and unlabeled webpages constitute a large part. Semi-supervised learning [2–4] is now well-known as a popular technique that exploits unlabeled data to help improve learning performance, particularly when there are limited labeled examples. During the past decade, semi-supervised learning has attracted significant attention in the machine learning community. In particular, three representative works on semi-supervised learning [5–7] have been awarded the 10-Year Best Paper by ICML in 2008, 2009, and 2013, respectively.

Among the many semi-supervised learning approaches [2,3], graph-based semi-supervised learning [1,7–9] is one of the most important semi-supervised learning paradigms. This type of methods is generally based on an assumption that similar input instances should be shared by similar output labels. It encodes both the labeled and unlabeled data as vertices in a weighted graph, with edge weights encoding the similarity between instances. The graph-based semi-supervised learning method aims to assign labels to unlabeled instances such that the inconsistency with respect to the graph is minimized. Because of its promising performance, the graph-based semi-supervised learning method has been widely employed in many application tasks, such as webpage classification [2] and image annotation [10].

Previous studies generally expected that when the amount of labeled data is limited, graph-based semi-supervised learning [1, 7, 11–13] could be an effective approach to improve the performance by exploiting auxiliary unlabeled data. However, in many cases [14–17], graph-based semi-supervised learning algorithms using auxiliary unlabeled data might even decrease the learning performance. Such situa-

Received November 17, 2016; accepted March 2, 2017

E-mail: {wanghai, wangsb, liyf}@lamda.nju.edu.cn

tions obviously affect the deployment of graph-based semi-supervised learning in many applications, especially those that require high reliability when using unlabeled data. For example, in the detection of VIP bank clients, only a small number of clients are known to be VIP or not; many clients are required to be tagged as latent VIP or not; incorrectly tagging clients will be an unacceptable disaster for the bank's service provision and final performance evaluation. To enable graph-based semi-supervised learning to be accepted by more users in practice, it is desirable to study the reduction of performance degeneration when using unlabeled data. This line of research is relatively young. Some recent efforts [18–20] have been devoted to this issue; however, they focus on semi-supervised SVMs [21, 22]. Efforts on graph-based semi-supervised learning in this context is quite limited [17].

In addition, a variety of graph-based semi-supervised learning methods [1, 7, 11, 23–26] have been proposed for transductive inference. However, these methods do not provide an out-of-sample extension, which means that we can only predict instances that are already observed in the graph during training. In certain application areas, it is desirable to learn a classifier such that predictions can be made on instances unobserved during training.

In this paper, based on our preliminary work [27], we propose an instance selection method in order to reduce the chances of performance degeneration when using unlabeled data, for both transductive and inductive situations. Our basic idea is that, rather than exploiting all unlabeled data, the unlabeled instances should be selected such that only the ones that are highly likely to be helpful are exploited, while the ones with high risks are excluded. To reduce the chances of performance degeneration, we first present two simple approaches based on predictive label aggregation and predictive confidence aggregation, respectively. By examining the limitations of these simple approaches, we propose our GssLIs (Graph Semi-Supervised Learning with Instance Selection) method which exploits both the predictive label and confidence simultaneously. We develop both transductive and inductive variants of our method. Extensive experiments on a broad range of data sets show that the chances of performance degeneration of our proposed method are much smaller than those of many state-of-the-art graph-based semi-supervised learning methods.

This paper is organized as follows. Section 2 briefly introduces the background of this work. Section 3 presents our GssLIs method. Experimental results are reported in Section 4. Section 5 concludes this paper.

2 Background

The key to graph-based semi-supervised learning is the prior data assumption that similar instances are likely to have similar labels. For simplicity of notation, the training data set is denoted as $D = \{\{\boldsymbol{x}_i, y_i\}_{i=1}^{l}, \{\boldsymbol{x}_j\}_{j=l+1}^{l+u}\}$ where $L = \{\boldsymbol{x}_i, y_i\}_{i=1}^{l}$ corresponding to the labeled instances and $U = \{\boldsymbol{x}_j\}_{j=l+1}^{l+u}$ corresponding to the unlabeled instances. $y_i \in \{+1, -1\}$ corresponds the label of instance $\boldsymbol{x}_i$, $i = 1, \ldots, l$. In graph-based semi-supervised learning, a graph $G(V, \boldsymbol{W})$ is constructed with nodes V corresponding to the $l + u$ training instances, with edges $\boldsymbol{W} = [w_{ij}] \in \mathcal{R}^{(l+u)\times(l+u)}$ corresponding to the weighted similarity matrix between the training instances. The graph-based semi-supervised learning method aims to assign class labels to unlabeled instances such that the label inconsistency with respect to the graph is minimized. Over the past decade, many graph-based semi-supervised learning methods [5–7, 23, 28] have been proposed. In this section, we briefly introduce two classical approaches: the Class Mass Normalization (CMN) method [7] and the Learning with Local and Global Consistency (LLGC) method [1].

2.1 Class mass normalization

CMN defines a function $f : L \cup U \to \mathcal{R}$ over the nodes. The label of a training instance is assigned based on the value of f. According to the assumption of graph-based semi-supervised learning, similar instances have similar labels, and this motivates CMN to select the quadratic energy function

$$E(f) = \frac{1}{2}\sum_{i,j=1}^{l+u} w_{ij}(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j))^2. \tag{1}$$

Let $\boldsymbol{D} \in \mathcal{R}^{(l+u)\times(l+u)}$ be a diagonal degree matrix, where $D_{ii} = \sum_{j=1}^{l+u} w_{ij}$ denotes the degree of instance $\boldsymbol{x}_i$ and the Laplacian matrix of graph $G(V, \boldsymbol{W})$ be $\boldsymbol{\Delta} = \boldsymbol{D} - \boldsymbol{W}$ [29]. Equation (1) can be rewritten in the following matrix-vector form

$$E(f) = \frac{1}{2}\sum_{i,j=1}^{l+u} w_{ij}(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j))^2 = \frac{1}{2}\boldsymbol{f}^{\mathrm{T}}\boldsymbol{\Delta}\boldsymbol{f}, \tag{2}$$

where $\boldsymbol{f} = [f(\boldsymbol{x}_1); \ldots; f(\boldsymbol{x}_{l+u})]$. To minimize Equation 2, since it is of a convex quadratic form, its optimal solution can be formulated as a closed-form solution,

$$\boldsymbol{f}_U = (-\boldsymbol{\Delta}_{UU})^{-1}\boldsymbol{\Delta}_{UL}\boldsymbol{y}_L, \tag{3}$$

where $\boldsymbol{f}_U = [f(\boldsymbol{x}_{l+1}); \ldots; f(\boldsymbol{x}_{l+u})]$ refers to the prediction on unlabeled data, $\boldsymbol{y}_L = [y_1; \ldots; y_l]$ is the ground-truth label on labeled data. The two matrices ($\mathbf{\Delta}_{UU}$ and $\mathbf{\Delta}_{UL}$) are partitioned from $\mathbf{\Delta}$,

$$\mathbf{\Delta} = \begin{bmatrix} \mathbf{\Delta}_{LL} & \mathbf{\Delta}_{LU} \\ \mathbf{\Delta}_{UL} & \mathbf{\Delta}_{UU} \end{bmatrix}. \tag{4}$$

Assume that the desirable proportions for classes “1” and “–1” are q and $1 - q$ respectively, where q is estimated from labeled data. Define the mass of class 1 to be $\sum_{i=l+1}^{l+u} f(\boldsymbol{x}_i)$ and the mass of class -1 to be $\sum_{i=l+1}^{l+u}(1 - f(\boldsymbol{x}_i))$. CMN scales these masses to match q and $1 - q$, respectively. Specifically, the prediction of an unlabeled instance $\boldsymbol{x}_j$ is defined as

$$\tilde{f}(\boldsymbol{x}_j) = q\frac{f(\boldsymbol{x}_j)}{\sum\limits_{i=l+1}^{l+u} f(\boldsymbol{x}_i)} - (1-q)\frac{1 - f(\boldsymbol{x}_j)}{\sum\limits_{i=l+1}^{l+u}(1 - f(\boldsymbol{x}_i))}. \tag{5}$$

Finally, the label of $\boldsymbol{x}_j$ is assigned as $\tilde{y}_j = sign(\tilde{f}(\boldsymbol{x}_j))$.

2.2 Learning with local and global consistency

The LLGC method considers a similar idea as CMN, but it considers the use of a matrix form rather than a vector form for the predictive result. Additionally, unlike the CMN method, which enforces that the prediction of graph-based semi-supervised learning on the labeled data must be the same as the ground-truth label, the LLGC method introduces a loss function for the labeled data, which allows some small losses on the labeled data. Specifically, the LLGC method considers optimizing the following objective,

$$\begin{aligned} E(f) = {} & \frac{\mu}{2}\sum_{i=1}^{l}(f(\boldsymbol{x}_i) - y_i)^2 \\ & + \frac{1}{2}\sum_{i,j=1}^{l+u} w_{ij}\Big(\frac{1}{\sqrt{D_{ii}}}f(\boldsymbol{x}_i) - \frac{1}{\sqrt{D_{jj}}}f(\boldsymbol{x}_j)\Big)^2, \end{aligned} \tag{6}$$

where μ trades off the relative importance between the labeled and unlabeled data. To optimize Equation 6, let $\mathcal{F}$ denote the set of $(l + u) \times 2$ matrices with nonnegative entries, where each element of these matrices is valued from $[0, 1]$, representing a confidence of the predictive result (the higher the value, the higher the confidence) and 2 represents that there are two classes. Let $\boldsymbol{F} = [F_{ij}] \in \mathcal{F}$ denote the classification on the entire training dataset D. Similar to the CMN method, the optimal $\boldsymbol{F}$ can be formulated as a closed-form solution,

$$\boldsymbol{F}^{\star} = (\boldsymbol{I} - \alpha\boldsymbol{S})^{-1}\boldsymbol{Y}, \tag{7}$$

where $\alpha = \frac{1}{1+\mu}$ (it is set to 0.99 in the LLGC method), $\boldsymbol{Y} \in \mathcal{F}$ with $Y_{i1} = 1, Y_{i2} = 0$ if $\boldsymbol{x}_i$ is labeled as 1; $Y_{i1} = 0, Y_{i2} = 1$ if $\boldsymbol{x}_i$ is labeled as 0; and $Y_{i1} = 0, Y_{i2} = 0$ otherwise; $\boldsymbol{S} = \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$ refers to a normalized Laplacian matrix. The final prediction of unlabeled instance $\boldsymbol{x}_j$ is computed as $f(\boldsymbol{x}_j) = (F^{\star}_{j1} - F^{\star}_{j2})/(F^{\star}_{j1} + F^{\star}_{j2})$ and the label $\tilde{y}_j$ is assigned as $sign(f(\boldsymbol{x}_j))$.

3 Our proposed method

Classical graph-based semi-supervised learning studies generally expected that when the amount of labeled data is limited, graph-based semi-supervised learning [1,7,11–13] could improve the performance by exploiting auxiliary unlabeled data. However, in many empirical cases [14–17], graph-based semi-supervised learning algorithms using auxiliary unlabeled data might even decrease the learning performance. To enable graph-based semi-supervised learning to be accepted by more users, in this section, we present ways to reduce the chances of performance degeneration when using unlabeled data in graph-based semi-supervised learning.

To alleviate this problem, we propose a new graph-based semi-supervised learning method GsslIs based on instance selection in order to reduce the chances of performance degeneration. Our basic idea is that given a set of unlabeled data, it is not the best approach to exploit all the unlabeled instances without any sanity check. Instead, we should exploit the unlabeled instances that are highly likely to help improve the performance, while not considering the ones with high risk. Based on this concept, in the following, we first present two simple approaches based on predictive label aggregation and predictive confidence aggregation, respectively. Then, by examining the limitations of these two simple approaches, we propose our GsslIs method that uses both the predictive label and confidence simultaneously, for both transductive and inductive situations.

3.1 Two simple approaches

3.1.1 MV

The first simple approach is the use of the majority voting (MV) strategy [30], which is known as an effective approach to improve the robustness of a learning method. In this study, it is used to aggregate multiple predictive labels from multiple graph-based semi-supervised learning methods (for example, through using multiple graphs). The label of an unlabeled instance is assigned as the majority one among multiple predictive labels. MV is simple and intuitively helpful to avoid the use of poor predictive labels. However, such a simple approach is not sufficient to derive a reliable result. Algorithm 1 provides the algorithm of the MV method.

Algorithm 1 Majority voting

```
Input: $L = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$, $U = \{\boldsymbol{x}_j\}_{j=l+1}^{l+u}$, a set of candidate weighted similarity matrices $\{\boldsymbol{W}_m\}_{m=1}^{M}$.
Output: A label assignment on training data $\tilde{\boldsymbol{y}} = [\tilde{y}_1, \ldots, \tilde{y}_{l+u}]$.
Perform the classical GSSL method (for example, CMN or LLGC) on a set of weighted similarity matrices $\{\boldsymbol{W}_m\}_{m=1}^{M}$, and collect the predictive value $\boldsymbol{F} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M]$ where
    $\boldsymbol{f}_m = [f^m(\boldsymbol{x}_1), \ldots, f^m(\boldsymbol{x}_{l+u})], \quad \forall m = 1, \ldots, M.$
Let
    $\mathcal{P} = \{i | sign(f^1(\boldsymbol{x}_i)) + \cdots + sign(f^M(\boldsymbol{x}_i)) \geqslant 0, \ i = 1, \ldots, l+u\},$
    $\mathcal{N} = \{i | sign(f^1(\boldsymbol{x}_i)) + \cdots + sign(f^M(\boldsymbol{x}_i)) < 0, \ i = 1, \ldots, l+u\}.$
For $\boldsymbol{x}_i \in L \cup U$:
    $\tilde{y}_i = \begin{cases} +1, & i \in \mathcal{P}; \\ -1, & i \in \mathcal{N}. \end{cases}$
return $\tilde{\boldsymbol{y}} = [\tilde{y}_1, \ldots, \tilde{y}_{l+u}]$.
```

3.1.2 DirA

Direct Aggregation (DirA) is motivated by predictive confidence aggregation, where the confidence obtained by the graph-based semi-supervised learning method can be regarded as a measurement of the reliability of unlabeled data. Formally, let $\boldsymbol{f}_m = [f^m(\boldsymbol{x}_1), \ldots, f^m(\boldsymbol{x}_{l+u})]$ denote the predictive value on a set of weight matrices $\{\boldsymbol{W}_m\}_{m=1}^{M}$ where M is the number of graphs. The DirA method aggregates the predictive values. The unlabeled instances with a high confidence value (or a high rank) are selected for use and the ones with a low confidence value (or a low rank) are risky and not exploited. DirA considers more detailed predictive values, rather than the labels themselves as in MV. Algorithm 2 provides the algorithm of the DirA method.

3.2 The GssLIs method

The use of both the MV and DirA methods has not been proposed before. Our empirical studies show that they both could reduce the chances of performance degeneration. However, they suffer from some limitations. The MV method considers only the hard label aggregation and may be risky when some hard labels are with low confidences. The DirA method considers only the mean of the predictive values whereas it ignores their variance, which might be misled and risky.

To alleviate the above deficiencies, in this paper we propose the GssLIs method. The basic observation is that the MV and DirA methods work complementarily to each other. Specifically, the predictive value aggregation used in the DirA method is able to avoid low confidence unlabeled instances, and thus could be applied to improve the MV method. On the other hand, the MV method proposes to use the unlabeled data with general consistent labels on multiple graphs, and this could consequently help exclude unlabeled data whose predictive values are with high variance.

Algorithm 2 Direct aggregation

```
Input: $L = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$, $U = \{\boldsymbol{x}_j\}_{j=l+1}^{l+u}$, a set of weighted similarity matrices $\{\boldsymbol{W}_m\}_{m=1}^{M}$, the predictive results of the inductive 1-nearest-neighbor algorithm $\hat{\boldsymbol{y}} = [\hat{y}_1, \ldots, \hat{y}_{l+u}]$ and parameter $\lambda$.
Output: A label assignment on training data $\tilde{\boldsymbol{y}} = [\tilde{y}_1, \ldots, \tilde{y}_{l+u}]$.
Perform the classical GSSL method (for example, CMN or LLGC) on a set of weighted similarity matrices $\{\boldsymbol{W}_m\}_{m=1}^{M}$, and collect the predictive value $\boldsymbol{F} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M]$ where
    $\boldsymbol{f}_m = [f^m(\boldsymbol{x}_1), \ldots, f^m(\boldsymbol{x}_{l+u})], \quad \forall m = 1, \ldots, M.$
For $\boldsymbol{x}_i \in L \cup U$, calculate the aggregated confidence $A_i$ according to the predictive values $[\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M]$
    $A_i = \frac{1}{M} \sum_{\mathrm{m}=1}^{M} f^m(\boldsymbol{x}_i).$
For $\boldsymbol{x}_i \in L \cup U$, assign predictive label $\tilde{y}_i$ according to $A_i$
    $\tilde{y}_i = \begin{cases} +1, & rank(A_i) \text{ (in a descending order)} \leqslant \frac{\lambda(l+u)l^+}{l}; \\ -1, & rank(A_i) \text{ (in an ascending order)} \leqslant \frac{\lambda(l+u)l^-}{l}; \\ \hat{y}_i, & \text{otherwise}, \end{cases}$
    where $l^+ (l^-)$ refers to the number of positive (negative) instances in labeled data.
return $\tilde{\boldsymbol{y}} = [\tilde{y}_1, \ldots, \tilde{y}_{l+u}]$.
```

Based on this observation, the GssLIs method is proposed. As Algorithm 3 shows, GssLIs first obtains the positive set $\mathcal{P}$ and the negative set $\mathcal{N}$ using the MV method, and then aggregates the predictive confidences on set $\mathcal{P}$ and $\mathcal{N}$, respectively. After aggregating the predictive confidences on $\mathcal{P}$ and $\mathcal{N}$, our proposal selects the unlabeled instances with high confidence while the unlabeled instances with low confidence are not selected. The time complexity of our method is $O(MT + (M+1)(l+u))$ where T is the time complexity of the GSSL method and M is the number of graphs, which is very small; therefore the proposal is as scalable as classical graph-based semi-supervised learning approaches. As will be shown in the experiment section, such an approach effectively improves the reliability of graph-based semi-supervised learning when using unlabeled data.

3.3 Out-of-sample extension

We consider transductive learning in the above formulation, which means we can only predict instances that are already observed in the graph at training time. However, in many cases, it is desirable to learn a classifier that can generalize to unobserved instances. Among the previously proposed graph-based semi-supervised learning methods which aim to assign labels to unlabeled instances such that the inconsistency with respect to the graph is minimized, this can be rewritten as the following form

$$E(f) = \frac{1}{2}\sum_{i,j=1}^{l+u} w_{ij}(f(\boldsymbol{x_i}) - f(\boldsymbol{x_j}))^2 + \beta\sum_{i=1}^{l}(f(\boldsymbol{x_i}) - y_i)^2, \quad (8)$$

where the hyper-parameter β controls the trade-off between those two costs.

```
Algorithm 3  The proposed method (transductive)
Input: $L = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$, $U = \{\boldsymbol{x}_j\}_{j=l+1}^{l+u}$, a set of weighted similarity matrices $\{\boldsymbol{W}_m\}_{m=1}^{M}$, the predictive results of the inductive 1-nearest-neighbor algorithm $\hat{\boldsymbol{y}} = [\hat{y}_1, \ldots, \hat{y}_{l+u}]$ and parameter $\lambda$;
Output: A label assignment on training data $\tilde{\boldsymbol{y}} = [\tilde{y}_1, \ldots, \tilde{y}_{l+u}]$.
Perform the classical GSSL method on a set of weighted similarity matrices $\{\boldsymbol{W}_m\}_{m=1}^{M}$, and collect the predictive value $\boldsymbol{F} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M]$ where
    $\boldsymbol{f}_m = [f^m(\boldsymbol{x}_1), \ldots, f^m(\boldsymbol{x}_{l+u})], \ \forall m = 1, \ldots, M.$
Let
    $\mathcal{P} = \{i | sign(f^1(\boldsymbol{x}_i)) + \cdots + sign(f^M(\boldsymbol{x}_i)) \geqslant 0, \ i = 1, \ldots, l+u\},$
    $\mathcal{N} = \{i | sign(f^1(\boldsymbol{x}_i)) + \cdots + sign(f^M(\boldsymbol{x}_i)) < 0, \ i = 1, \ldots, l+u\}.$
For $\boldsymbol{x}_i \in L \cup U$, calculate the aggregated confidence $A_i$ according to the predictive values $[\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M]$
    $A_i = \frac{1}{M}\sum_{m=1}^{M} f^m(\boldsymbol{x}_i).$
For $\boldsymbol{x}_i \in L \cup U$, assign predictive label $\tilde{y}_i$ according to $A_i$
    $\tilde{y}_i = \begin{cases} +1, & i \in \mathcal{P} \ \& \ rank(A_i) \text{ (in a descending order)} \leqslant \lambda|\mathcal{P}|; \\ -1, & i \in \mathcal{N} \ \& \ rank(A_i) \text{ (in an ascending order)} \leqslant \lambda|\mathcal{N}|; \\ \hat{y}_i, & \text{otherwise.} \end{cases}$
return $\tilde{\boldsymbol{y}} = [\tilde{y}_1, \ldots, \tilde{y}_{l+u}]$.
```

Let $\boldsymbol{A} = \beta\boldsymbol{\Delta}_L + \boldsymbol{\Delta}$, where $(\boldsymbol{\Delta}_L)_{ij} = \delta_{ij}\delta_{i\in L}$ and $\boldsymbol{\Delta}$ denote the Laplacian matrix of graph $G(V, \boldsymbol{W})$ [31]. Equation (8) can be rewritten as the following linear system

$$\boldsymbol{A}\boldsymbol{f} = \beta\boldsymbol{y}. \quad (9)$$

To obtain function induction without having to solve linear system Eq. (9) for each new test data, motivated by [31], an inductive variant of GssLIs is proposed. Considering the smoothness criterion in graph-based semi-supervised learning and keeping the value of $f(\boldsymbol{x_j})$ fixed on the training points $\boldsymbol{x_j}$ lead to the minimization of Eq. (1). Then, we can obtain the solution as the following form

$$f(\boldsymbol{x_i}) = \frac{\sum\limits_{j=1}^{l+u} w_{ij} f(\boldsymbol{x_j})}{\sum\limits_{j=1}^{l+u} w_{ij}}. \quad (10)$$

This induction formula gives the same result as the transduction formula over unlabeled data. The estimated $f(\boldsymbol{x_j})$ on the training set can be obtained from Eq. (9). For a new example $\boldsymbol{x_i}$, we can compute its label $f(\boldsymbol{x_i})$ using Eq. (10). Algorithm 4 gives the inductive variant of the GssLIs algorithm. Similar approaches can be used to derive the inductive variant of the MV and DirA methods.

```
Algorithm 4  The proposed method (inductive)
Input: Training data set $D = \{\{\boldsymbol{x}_i, y_i\}_{i=1}^{l}, \{\boldsymbol{x}_j\}_{j=l+1}^{l+u}\}$ and test data set $T = \{\{\boldsymbol{x}_j\}_{j=l+u+1}^{z}\}$, a set of weighted similarity matrices $\{\boldsymbol{W}_m\}_{m=1}^{M}$ on training data set $D$, the predictive results of the inductive 1-nearest-neighbor algorithm $\hat{\boldsymbol{y}} = [\hat{y}_{l+u+1}, \ldots, \hat{y}_z]$ and parameter $\lambda$;
Output: A label assignment on test instances $\tilde{\boldsymbol{y}} = [\tilde{y}_{l+u+1}, \ldots, \tilde{y}_z]$.
Perform the classical GSSL methods on a set of weighted similarity matrices $\{\boldsymbol{W}_m\}_{m=1}^{M}$, and collect the predictive value $\boldsymbol{F} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M]$ where
    $\boldsymbol{f}_m = [f^m(\boldsymbol{x}_1), \ldots, f^m(\boldsymbol{x}_{l+u})], \ \forall m = 1, \ldots, M.$
For $\boldsymbol{x}_i \in T$, compute its label $f(\boldsymbol{x_i})$ by
    $f^m(\boldsymbol{x_i}) = \frac{\sum_{j=1}^{l+u} w_{ij} f^m(\boldsymbol{x_j})}{\sum_{j=1}^{l+u} w_{ij}}, \ \forall m = 1, \ldots, M.$
Let
    $\mathcal{P} = \{i | sign(f^1(\boldsymbol{x}_i)) + \cdots + sign(f^M(\boldsymbol{x}_i)) \geqslant 0, \ i = l+u+1, \ldots, z\},$
    $\mathcal{N} = \{i | sign(f^1(\boldsymbol{x}_i)) + \cdots + sign(f^M(\boldsymbol{x}_i)) < 0, \ i = l+u+1, \ldots, z\}.$
For $\boldsymbol{x}_i \in T$, calculate the aggregated confidence $A_i$ according to the predictive values $[\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M]$,
    $A_i = \frac{1}{M}\sum_{m=1}^{M} f^m(\boldsymbol{x}_i).$
For $\boldsymbol{x}_i \in T$, assign predictive label $\tilde{y}_i$ according to $A_i$
    $\tilde{y}_i = \begin{cases} +1, & i \in \mathcal{P} \ \& \ rank(A_i) \text{ (in a descending order)} \leqslant \lambda|\mathcal{P}|; \\ -1, & i \in \mathcal{N} \ \& \ rank(A_i) \text{ (in an ascending order)} \leqslant \lambda|\mathcal{N}|; \\ \hat{y}_i, & \text{otherwise.} \end{cases}$
return $\tilde{\boldsymbol{y}} = [\tilde{y}_{l+u+1}, \ldots, \tilde{y}_z]$.
```

3.4 Discussion

It should be noted that the basic idea of this paper is not specific to graph-based semi-supervised learning (a similar idea has been applied to semi-supervised SVM [32] and achieved promising performance). Moreover, it is worth noting that our idea may also be applicable to improving the performance of semi-supervised ranking problems (e.g., [33]) by excluding highly risky unlabeled data that are sensitive to predict the ranking scores of neighboring points. In this paper, we focus on the pure semi-supervised learning setting and the extension of active learning (e.g., a nice literature [34]) is interesting, however, beyond the scope of this paper. An interesting benefit of our paper to active learning is that it might provide a way of selecting uncertain, diverse, and reliable unlabeled data from the active pool.

4 Experiments

In this section, we first describe the data sets and the compared methods. Then, we present the experimental results mainly in three aspects, i.e., comparison results, comparison results with improved baseline, and influence on the number of graphs.

4.1 Data sets

Sixteen UCI data sets are employed to conduct the experiments (Table 1). For transductive graph-based semi-supervised learning, in each data set, ten examples are randomly chosen as the labeled examples, and the remaining data are used as the unlabeled data. We evaluate the performance of each method on the unlabeled data. For inductive graph-based semi-supervised learning, we randomly choose 50% of the available data as the training set; ten examples are randomly chosen as the labeled examples in the training set, and the remaining data are used as the test set. We evaluate the performance of each method on the test set. The experiments are repeated 30 times and the average accuracies with their standard deviations are recorded.

Table 1 Data sets used in experiment

Data	# Dim.	# Positive	# Negative	# Total
Text	11,960	750	750	1,500
Credit-approval	15	383	307	690
Hill-valley	100	606	606	1,212
Breastw	9	239	444	683
House-votes	16	267	168	435
Digit1	241	734	766	1,500
Wdbc	14	357	212	569
Isolet	51	300	300	600
Liver disorders	6	200	145	345
Spambase	57	1,813	2,788	4,601
Vehicle	16	218	217	435
Statlog-heart	13	120	150	270
House	16	108	124	232
German	24	300	700	1,000
Diabetes	8	500	268	768
Horse-colic	25	136	232	368

4.2 Compared methods

The proposal is compared with the following methods.

- 1NN: Classical supervised 1-nearest-neighbor method, which is used as a baseline supervised approach. In this approach, each unlabeled instance is assigned the label of its nearest labeled example [1].
- CMN: Class Mass Normalization method proposed by [7], as mentioned in Section 2.1.
- LLGC: Learning with Local and Global Consistency method proposed by [1], as mentioned in Section 2.2.
- MV: Majority Voting method in Section 3.1.1.
- DirA: Direct Aggregation method in Section 3.1.2.

The parameter setups for the compared methods are as follows. For the 1NN method, the Euclidean distance metric is used to locate the nearest neighbors. For CMN and LLGC, 5-nearest-neighbor graphs under three kinds of distance metrics (namely, Euclidean distance, Cosine distance, and Manhattan distance) are used for comparison. The parameter of the CMN method is set to the recommended one in the package. The LLGC method is implemented by ourselves and the parameter α is set to 0.99 as recommended in the paper. For MV, DirA, and our proposed GssLIs method, the 5-nearest-neighbor graphs with three kinds of distance metrics are employed as the set of graphs. The parameter λ in DirA is set to 0.5 (we have tried other parameters and found that 0.5 works the best). In our proposed GsslIs method, the parameter λ is set to 0.7 for all the experimental cases. As for the inductive variant of GssLIs, the parameter β is set to 100 as recommended in the paper [31].

4.3 Comparison results

Among the transductive methods, Table 2 shows the comparison results based on the implementation of CMN, and Table 3 shows the comparison results based on the implementation of LLGC. As can be seen in Table 2, our method GssLIs achieves highly competitive performance with the compared methods. For example, in terms of average accuracy, GssLIs obtains the best average accuracy. More importantly, all the compared graph-based semi-supervised learning methods will significantly decrease the performance in many cases, whereas our proposed approach never degenerates the performance. Although both the MV and DirA methods are capable of reducing the chances of performance degeneration, they still degenerate the performance in multiple cases, whereas our proposed method does not exhibit such kind of phenomena.

As for the LLGC method, Table 3 shows the comparison results. As can be seen, similar to the cases in Table 2, GssLIs also obtains highly competitive performance with the compared graph-based semi-supervised methods, whereas unlike the compared methods which significantly degenerate the performance in many cases, our proposed method does not degenerate the performance.

Table 2 Accuracy (mean ± std) on ten labeled instances for the compared methods based on CMN implementation

Data	1NN	CMN			MV	DirA	GssLIs
	Euclidean	Euclidean	Cosine	Manhattan			
Text	59.5±3.4	**64.0±4.8**	**63.1±4.6**	51.2±3.2	**64.2±4.9**	**62.9±4.2**	**64.5±4.5**
Credit	72.9±6.9	69.9±8.1	68.2±7.1	69.1±8.5	69.6±7.7	73.4±7.3	73.3±6.9
Hill	50.1±1.6	50.0±1.7	**66.9±5.7**	50.0±1.9	**51.8±2.5**	**50.6±1.7**	**50.9±1.6**
Breastw	93.2±3.6	**95.6±1.0**	73.1±4.4	**95.7±0.9**	**95.5±1.1**	93.2±3.4	**93.5±3.1**
House-v	86.7±3.0	**88.7±1.8**	87.7±3.2	**88.7±2.4**	**87.9±2.3**	**87.1±2.4**	**87.5±2.0**
Digit1	78.1±5.3	**86.2±3.5**	**85.3±4.0**	**83.3±3.7**	**86.8±3.3**	**80.8±4.3**	**84.0±3.8**
Wdbc	80.5±5.5	79.8±4.6	77.3±5.2	73.8±4.3	78.6±5.3	**82.9±4.8**	**85.1±3.8**
Isolet	91.6±3.6	**98.0±0.9**	**98.4±0.8**	**97.7±1.1**	**98.5±0.7**	**92.2±3.4**	**93.2±2.8**
Liver	52.6±3.2	52.0±3.3	53.1±4.8	52.4±3.0	52.8±4.1	52.7±3.0	53.2±3.2
Spambase	69.4±8.0	61.5±1.6	61.7±1.2	61.6±1.1	61.4±1.2	**73.9±6.4**	**71.3±7.0**
Vehicle	72.8±6.0	**74.4±7.4**	**78.1±9.0**	**79.5±8.5**	**78.9±8.7**	**74.2±6.0**	**76.3±7.1**
Statlog	73.3±5.9	74.1±5.8	74.0±4.8	**77.0±5.2**	**77.0±4.6**	**75.2±4.5**	**76.7±4.0**
House	89.4±2.1	89.8±2.2	88.5±2.8	88.0±2.1	89.6±2.1	89.4±2.1	89.6±1.9
German	63.8±5.2	**69.0±1.3**	**69.3±1.3**	**69.7±0.8**	**69.4±1.0**	62.5±4.5	**65.8±3.0**
Diabetes	64.5±5.3	65.6±2.1	66.5±2.0	65.5±2.5	65.7±2.0	64.3±5.2	**66.0±3.3**
Horse	65.3±4.6	65.1±4.6	66.6±5.6	64.8±4.0	65.8±5.1	65.6±4.7	**68.1±5.2**
Ave.Acc.	72.7	74.0	73.6	73.0	74.6	73.8	74.9
W/T/L against 1NN		7/7/2	6/6/4	7/4/5	9/4/3	9/6/1	**13/3/0**

Note: For the graph-based semi-supervised learning methods, if the performance is significantly better/worse than 1NN, the corresponding entries are bolded/underlined (paired t-tests at 95% significance level). The average accuracy is listed for comparison. The win/tie/loss counts are summarized and the method with the smallest number of losses against 1NN is bolded

Table 3 Accuracy (mean ± std) on ten labeled instances for the compared methods based on LLGC implementation

Data	1NN	LLGC			MV	DirA	GssLIs
	Euclidean	Euclidean	Cosine	Manhattan			
Text	59.5±3.4	55.8±4.8	57.6±5.9	50.7±1.2	56.2±5.4	**62.9±4.4**	**61.0±4.9**
Credit	72.9±6.9	69.9±7.6	68.8±7.7	69.6±8.7	68.8±7.8	73.3±7.4	73.0±6.9
Hill	50.1±1.6	50.0±1.8	**68.5±6.3**	50.0±1.7	**50.6±1.7**	**51.7±2.2**	**51.3±1.9**
Breastw	93.2±3.6	**95.8±0.6**	78.2±7.0	**95.9±0.6**	**95.6±0.5**	93.3±3.4	**93.6±2.9**
House-v	86.7±3.0	82.9±8.1	84.7±6.5	84.6±7.2	84.9±7.0	**87.2±2.4**	**87.2±3.0**
Digit1	78.1±5.3	**90.0±3.4**	**90.1±2.8**	**88.0±3.1**	**90.9±3.0**	**80.7±4.4**	**84.0±3.7**
Wdbc	80.5±5.5	71.0±6.8	70.9±6.1	67.1±3.7	69.2±4.9	**83.0±4.7**	**85.4±4.2**
Isolet	91.6±3.6	**97.0±1.9**	**97.9±0.8**	**97.2±1.8**	**98.3±1.1**	**92.1±3.5**	**93.3±2.7**
Liver	52.6±3.2	52.5±3.8	53.9±4.7	51.9±4.8	53.2±4.1	53.0±2.8	53.0±3.2
Spambase	69.4±8.0	65.6±4.7	66.3±4.6	64.5±3.3	65.0±4.0	**74.0±6.1**	**72.0±6.8**
Vehicle	72.8±6.0	**74.8±7.9**	**77.1±8.5**	**80.2±8.5**	**77.2±8.4**	**73.9±5.9**	**75.4±6.8**
Statlog	73.3±5.9	60.6±5.7	59.2±4.7	59.2±5.3	59.7±5.7	**75.3±4.4**	**75.1±5.2**
House	89.4±2.1	80.8±11.7	83.3±8.4	79.9±11.3	82.4±10.3	89.4±2.1	88.8±2.7
German	63.8±5.2	**69.2±1.4**	**69.0±1.5**	**69.6±1.0**	**69.4±1.2**	62.6±4.4	**65.7±2.9**
Diabetes	64.5±5.3	65.4±2.1	66.0±1.9	65.4±1.8	65.5±1.7	64.3±5.2	**65.8±3.3**
Horse	65.3±4.6	63.0±3.4	63.5±4.0	62.6±3.1	62.4±2.4	65.4±4.8	**68.6±5.4**
Ave.Acc.	72.7	71.5	72.2	71.0	71.8	73.9	74.6
W/T/L against 1NN		5/3/8	5/5/6	5/4/7	6/3/7	9/6/1	**13/3/0**

Note: Please refer to Note of Table 2

For out-of-sample (or inductive) graph-based semi-supervised learning methods, Table 4 shows the comparison results. As can be seen, unlike the MV and DirA methods, which significantly degenerate the performance in many cases, our proposed method GssLIs does not degenerate the performance and also obtains the best average accuracy.

Overall, these results show that both the transductive and inductive variants of our proposed method are able to reduce the chances of performance degeneration, while still obtaining highly competitive performance improvement with the state-of-the-art graph-based semi-supervised learning methods.

4.4 Comparison results with improved baseline

We further consider the comparison results with the improved baseline approach. We implement the MV strategy for three 1NN baseline with three different distance metrics (namely, Euclidean distance, Cosine distance, and Manhattan distance). For transductive graph-based semi-supervised learning methods, Table 6 shows the comparison results in terms of CMN. As can be seen, the baseline supervised approach is improved. Even in these cases, our proposed method works. As shown in Table 6, similar observations can be found when the LLGC method is compared. For the inductive graph-based semi-supervised learning method, the out-of-sample extension method does not degenerate the performance whereas other graph-based semi-supervised learning methods significantly degenerate the performance in many

Table 4 Accuracy (mean ± std) on ten labeled instances for compared methods based on out-of-sample (inductive) extension

Data	1NN	Inductive			MV	DirA	GssLIs
	Euclidean	Euclidean	Cosine	Manhattan			
Text	60.1±5.3	53.4±5.2	53.4±5.2	50.5±1.6	53.4±5.2	**62.8±5.2**	59.2±5.0
Credit	71.6±5.1	70.1±6.7	70.3±6.7	70.1±6.6	70.2±6.7	69.6±4.9	72.1±5.2
Hill	50.0±2.1	49.3±2.0	50.2±2.2	49.4±1.9	49.4±2.0	50.0±2.2	50.0±2.0
Breastw	94.2±2.8	**95.8±1.6**	91.7±1.4	**95.8±1.4**	**95.9±1.4**	94.0±2.8	94.0±2.2
House-v	86.7±4.0	84.2±7.7	83.9±7.5	84.7±8.1	84.3±7.7	86.1±1.1	**87.2±3.5**
Digit1	78.6±5.7	**90.8±5.8**	**90.8±5.8**	**90.7±5.8**	**90.8±5.8**	**86.0±6.3**	**85.2±4.0**
Wdbc	80.1±6.0	69.5±6.3	69.5±6.1	68.6±5.4	69.3±6.2	**82.5±3.0**	**82.3±6.0**
Isolet	91.7±3.9	**96.3±5.2**	**96.6±5.1**	**96.3±5.1**	**96.6±5.1**	**94.6±2.4**	**93.1±3.3**
Liver	55.0±4.8	**56.8±3.2**	**57.8±3.1**	**57.6±3.2**	**57.6±2.9**	52.1±5.0	**56.0±4.3**
Spambase	69.7±8.0	62.1±2.9	62.3±3.0	62.1±2.9	62.0±2.9	**73.8±5.7**	**73.0±6.4**
Vehicle	72.6±6.6	70.8±7.6	72.5±8.1	71.4±7.6	71.2±7.8	72.7±8.5	73.5±6.9
Statlog	73.3±6.4	**77.3±6.2**	**77.4±6.2**	**77.3±6.0**	**77.7±6.3**	**76.5±6.0**	**76.4±5.7**
House	88.0±2.7	**89.1±2.4**	**89.5±2.2**	**89.1±2.4**	**89.1±2.4**	**89.4±2.4**	**88.5±2.3**
German	62.5±5.8	**69.5±1.6**	**69.5±1.6**	**69.6±1.6**	**69.5±1.6**	**63.6±4.5**	**64.9±3.5**
Diabetes	65.2±3.9	65.6±1.3	65.6±1.1	65.5±1.3	65.6±1.2	59.9±6.6	65.8±3.1
Horse	62.8±6.1	**64.9±7.6**	**64.5±6.7**	**65.2±6.9**	**65.0±7.3**	**64.5±6.3**	63.6±7.8
Ave.Acc.	72.6	72.8	72.8	72.7	73.0	73.6	74.0
W/T/L against 1NN		8/4/4	7/4/5	8/4/4	8/4/4	9/3/4	**9/7/0**

Note: Please refer to Note of Table 2

Table 5 Accuracy (mean ± std) with improved 1NN method (using MV strategy) based on CMN method

Data	1NN	CMN			MV	DirA	GssLIs
	Euclidean	Euclidean	Cosine	Manhattan			
Text	59.4±3.4	**64.0±4.8**	**63.1±4.6**	51.2±3.2	**64.2±4.9**	**62.9±4.2**	**64.5±4.5**
Credit	73.7±6.8	69.9±8.1	68.2±7.1	69.1±8.5	69.6±7.7	73.9±7.3	73.8±6.8
Hill	50.1±1.6	50.0±1.7	**66.9±5.7**	50.0±1.9	**51.8±2.5**	**50.6±1.7**	**50.8±1.6**
Breastw	93.4±3.2	**95.6±1.0**	73.1±4.4	**95.7±0.9**	**95.5±1.1**	93.4±3.0	**93.7±2.8**
House-v	87.2±2.7	**88.7±1.8**	87.7±3.2	**88.7±2.4**	87.9±2.3	87.5±2.2	**87.8±1.8**
Digit1	78.3±5.4	**86.2±3.5**	**85.3±4.0**	**83.3±3.7**	**86.8±3.3**	**81.0±4.3**	**84.0±3.8**
Wdbc	80.9±5.6	79.8±4.6	77.3±5.2	73.8±4.3	78.6±5.3	**83.3±4.9**	**85.3±3.9**
Isolet	92.0±3.6	**98.0±0.9**	**98.4±0.8**	**97.7±1.1**	**98.5±0.7**	**92.5±3.3**	**93.5±2.7**
Liver	52.8±3.1	52.0±3.3	53.1±4.8	52.4±3.0	52.8±4.1	52.9±3.1	53.3±3.1
Spambase	70.8±7.9	61.5±1.6	61.7±1.2	61.6±1.1	61.4±1.2	**74.5±6.6**	71.8±6.7
Vehicle	74.1±6.5	74.4±7.4	**78.1±9.0**	**79.5±8.5**	**78.9±8.7**	**75.5±6.5**	**77.1±7.3**
Statlog	74.1±5.4	74.1±5.8	74.0±4.8	**77.0±5.2**	**77.0±4.6**	**75.7±4.2**	**77.0±3.8**
House	89.4±2.1	89.8±2.2	88.5±2.8	88.0±2.1	89.6±2.1	89.4±2.1	89.6±1.9
German	63.8±5.1	**69.0±1.3**	**69.3±1.3**	**69.7±0.8**	**69.4±1.0**	62.5±4.4	**65.8±3.0**
Diabetes	64.9±5.3	65.6±2.1	66.5±2.0	65.5±2.5	65.7±2.0	64.7±5.3	**66.2±3.2**
Horse	65.7±5.1	65.1±4.6	66.6±5.6	64.8±4.0	65.8±5.1	65.9±5.2	**68.1±5.6**
Ave.Acc.	73.2	74.0	73.6	73.0	74.6	74.1	75.1
W/T/L against 1NN		6/8/2	6/6/4	7/4/5	8/5/3	8/7/1	**12/4/0**

Note: Please refer to Note of Table 2

Table 6 Accuracy (mean ± std) with improved 1NN method (using MV strategy) based on LLGC method

Data	1NN	LLGC			MV	DirA	GssLIs
	Euclidean	Euclidean	Cosine	Manhattan			
Text	59.4±3.4	55.8±4.8	57.6±5.9	50.7±1.2	56.2±5.4	**62.9±4.4**	**61.0±4.9**
Credit	73.7±6.8	69.9±7.6	68.8±7.7	69.6±8.7	68.8±7.8	73.9±7.3	73.5±6.7
Hill	50.1±1.6	50.0±1.8	**68.5±6.3**	50.0±1.7	**50.6±1.7**	**51.7±2.1**	**51.3±1.9**
Breastw	93.4±3.2	**95.8±0.6**	78.2±7.0	**95.9±0.6**	**95.6±0.5**	93.5±3.1	93.7±2.6
House-v	87.2±2.7	82.9±8.1	84.7±6.5	84.6±7.2	84.9±7.0	87.6±2.0	87.5±2.7
Digit1	78.3±5.4	**90.0±3.4**	**90.1±2.8**	**88.0±3.1**	**90.9±3.0**	**80.9±4.5**	**84.1±3.7**
Wdbc	80.9±5.6	71.0±6.8	70.9±6.1	67.1±3.7	69.2±4.9	**83.4±4.7**	**85.5±4.4**
Isolet	92.0±3.6	**97.0±1.9**	**97.9±0.8**	**97.2±1.8**	**98.3±1.1**	**92.5±3.4**	**93.6±2.7**
Liver	52.8±3.1	52.5±3.8	53.9±4.7	51.9±4.8	53.2±4.1	53.1±2.8	53.1±3.4
Spambase	70.8±7.9	65.6±4.7	66.3±4.6	64.5±3.3	65.0±4.0	**74.6±6.3**	**72.4±6.6**
Vehicle	74.1±6.5	74.8±7.9	**77.1±8.5**	**80.2±8.5**	**77.2±8.4**	**75.2±6.3**	**76.3±7.0**
Statlog	74.1±5.4	60.6±5.7	59.2±4.7	59.2±5.3	59.7±5.7	**75.8±4.2**	**75.7±4.7**
House	89.4±2.1	80.8±11.7	83.3±8.4	79.9±11.3	82.4±10.3	89.4±2.1	88.8±2.7
German	63.8±5.1	**69.2±1.4**	**69.0±1.5**	**69.6±1.0**	**69.4±1.2**	62.6±4.3	**65.7±2.8**
Diabetes	64.9±5.3	65.4±2.1	66.0±1.9	65.4±1.8	65.5±1.7	64.7±5.3	66.1±3.2
Horse	65.7±5.1	63.0±3.4	63.5±4.0	62.6±3.1	62.4±2.4	65.6±5.2	**68.6±5.7**
Ave.Acc.	73.2	71.5	72.2	71.0	71.8	74.2	74.8
W/T/L against 1NN		4/4/8	5/3/8	5/3/8	6/3/7	8/7/1	**10/6/0**

Note: Please refer to Note of Table 2

Table 7 Accuracy (mean ± std) with improved 1NN method for out-of-sample (inductive) extension

Data	1NN	Inductive			MV	DirA	GssLIs
	Euclidean	Euclidean	Cosine	Manhattan			
Text	60.0±5.3	53.4±5.2	53.4±5.2	50.5±1.6	53.4±5.2	**62.8±5.2**	59.2±5.0
Credit	72.8±5.0	70.1±6.7	70.3±6.7	70.1±6.6	70.2±6.7	70.0±4.7	72.7±5.1
Hill	50.0±2.1	49.3±2.0	50.2±2.2	49.4±1.9	49.4±2.0	49.9±2.2	49.9±2.0
Breastw	94.2±2.7	**95.8±1.6**	91.7±1.4	**95.8±1.4**	**95.9±1.4**	94.0±2.7	94.0±2.1
House-v	87.1±4.2	84.2±7.7	83.9±7.5	84.7±8.1	84.3±7.7	86.2±1.1	87.5±3.7
Digit1	78.8±5.6	**90.8±5.8**	**90.8±5.8**	**90.7±5.8**	**90.8±5.8**	**85.9±6.3**	**85.2±4.0**
Wdbc	80.4±6.3	69.5±6.3	69.5±6.1	68.6±5.4	69.3±6.2	**82.6±2.8**	**82.3±6.2**
Isolet	91.6±4.2	**96.3±5.2**	**96.6±5.1**	**96.3±5.1**	**96.6±5.1**	**95.4±2.2**	**93.2±3.5**
Liver	55.6±5.3	56.8±3.2	**57.8±3.1**	**57.6±3.2**	**57.6±2.9**	52.3±5.0	56.4±4.5
Spambase	72.0±7.8	62.1±2.9	62.3±3.0	62.1±2.9	62.0±2.9	**74.6±5.9**	73.7±5.9
Vehicle	74.0±7.0	70.8±7.6	72.5±8.1	71.4±7.6	71.2±7.8	73.1±8.6	74.5±6.9
Statlog	74.2±6.3	**77.3±6.2**	**77.4±6.2**	**77.3±6.0**	**77.7±6.3**	**76.8±6.0**	**76.8±5.7**
House	88.0±2.7	**89.1±2.4**	**89.5±2.2**	**89.1±2.4**	**89.1±2.4**	**89.4±2.4**	**88.5±2.3**
German	62.8±5.6	**69.5±1.6**	**69.5±1.6**	**69.6±1.6**	**69.5±1.6**	**63.8±4.4**	**65.0±3.5**
Diabetes	65.8±4.0	65.6±1.3	65.6±1.1	65.5±1.3	65.6±1.2	60.1±6.6	66.0±3.2
Horse	63.1±6.0	**64.9±7.6**	64.5±6.7	**65.2±6.9**	**65.0±7.3**	**64.8±6.2**	63.9±7.6
Ave.Acc.	73.2	72.8	72.8	72.7	73.0	73.9	74.3
W/T/L against 1NN		7/2/7	6/4/6	8/2/6	8/2/6	9/3/4	**6/10/0**

Note: Please refer to Note of Table 2

cases.

4.5 Influence on the number of graphs

We further study the influence on the number of candidate graphs. We generate the candidate graphs as follows. For each instance, the number of nearest neighbors is randomly selected between 3 to 7 with a uniform distribution. The number of candidate graphs $M \in \{3, 5, 7\}$ and the graph is constructed by using different distance metrics. Figures 1–3 show the results with different numbers of candidate graphs. As can be seen, for both transductive and inductive methods, our GssLIs method always improves the performance, and rarely degenerates the performance as the number of candidate graphs varies.

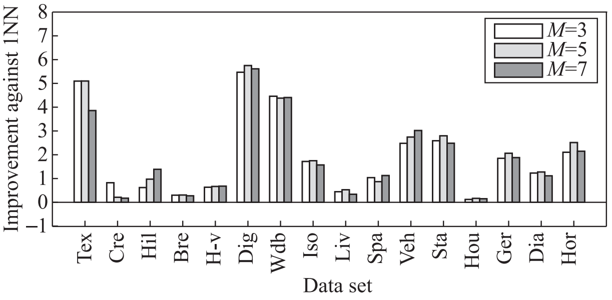


Fig. 1 Influence on the number of graphs M w.r.t. the improvement of GssLIs (based on CMN) against 1NN

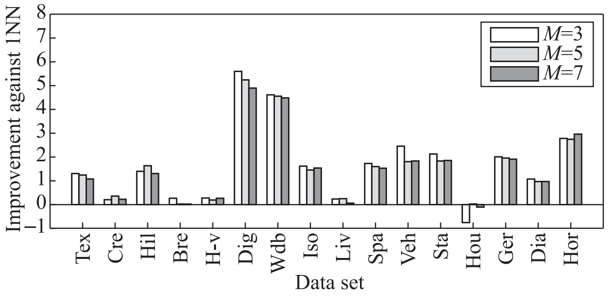


Fig. 2 Influence on the number of graphs M w.r.t. the improvement of GssLIs (based on LLGC) against 1NN

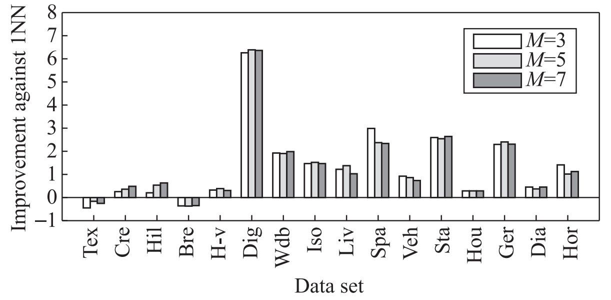


Fig. 3 Influence on the number of graphs M w.r.t. the improvement of GssLIs (out-of-sample extension) against 1NN

5 Conclusion

In this paper, we proposed a new graph-based semi-supervised learning method GssLIs based on instance selection in order to reduce the chances of performance degeneration. Our basic idea is that given a set of unlabeled instances, it is not the best approach to exploit all the unlabeled instances; instead, we should exploit the unlabeled instances that are highly likely to help improve the performance, while not considering the ones with high risk. We develop both transductive and inductive variants of our method. Experiments on a broad range of data sets show that the chances of performance degeneration of our proposed method are much smaller than those of many state-of-the-art graph-based semi-supervised learning methods.

Acknowledgements The authors want to thank the associate editors and reviewers for helpful comments and suggestions. This research was partially supported by the National Natural Science Foundation of China (Grant No. 61403186), Jiangsu Science Foundation (BK20140613) and MSRA research fund.

References

1. Zhou D, Bousquet O, Lal T N, Weston J, Schölkopf B. Learning with local and global consistency. In: Proceedings of the 16th International Conference on Neural Information Processing Systems. 2004, 321–328
2. Zhu X. Semi-supervised learning literature survey. Technical Report, 2007
3. Zhu X, Goldberg A B. Introduction to semi-supervised learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 2009, 3(1): 1–130
4. Chapelle O, Schölkopf B, Zien A. Semi-Supervised Learning. Cambridge: MIT Press, 2006
5. Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th Annual Conference on Computational Learning Theory. 1998, 92–100
6. Joachims T. Transductive inference for text classification using support vector machines. In: Proceedings of the 16th International Conference on Machine Learning. 1999, 200–209
7. Zhu X, Ghahramani Z, Lafferty J. Semi-supervised learning using Gaussian fields and harmonic functions. In: Proceedings of the 20th International Conference on Machine learning. 2003, 912–919
8. Zhu X, Lafferty J, Rosenfeld R. Semi-supervised learning with graphs. Dissertation for the Doctoral Degree. Pittsburgh: Carnegie Mellon University, 2005
9. Cai X F, Wen G H, Wei J, Yu Z W. Relative manifold based semi-supervised dimensionality reduction. Frontiers of Computer Science, 2014, 8(6): 923–932
10. Liu W, Wang J, Chang S F. Robust and scalable graph-based semisupervised learning. Proceedings of the IEEE, 2012, 100(9): 2624–2638
11. Joachims T. Transductive learning via spectral graph partitioning. In: Proceedings of the 20th International Conference on Machine Learning. 2003, 290–297
12. Zha Z J, Mei T, Wang J, Wang Z, Hua X S. Graph-based semi-supervised learning with multiple labels. Journal of Visual Communication and Image Representation, 2009, 20(2): 97–103
13. Camps-Valls G, Marsheva T V B, Zhou D. Semi-supervised graph-based hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 2007, 45(10): 3044–3054
14. Belkin M, Niyogi P. Semi-supervised learning on riemannian manifolds. Machine Learning, 2004, 56(1–3): 209–239
15. Karlen M, Weston J, Erkan A, Collobert R. Large scale manifold transduction. In: Proceedings of the 25th International Conference on Machine Learning. 2008, 775–782
16. Wang F, Zhang C. Label propagation through linear neighborhoods. IEEE Transactions on Knowledge and Data Engineering, 2008, 20(1): 55–67
17. Li Y F, Wang S B, Zhou Z H. Graph quality judgement: a large margin expedition. In: Proceedings of the 25th International Joint Confernece on Artificial Intelligence. 2016, 1725–1731

18. Li Y F, Zhou Z H. Towards making unlabeled data never hurt. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(1): 175–188

19. Li Y F, Kwok J T, Zhou Z H. Towards safe semi-supervised learning for multivariate performance measures. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence. 2016, 1816–1822

20. Balsubramani A, Freund Y. Optimally combining classifiers using unlabeled data. In: Proceedings of the 28th International Conference on Learning Theory. 2015, 211–225

21. Bennett K P, Demiriz A. Semi-supervised support vector machines. In: Proceedings of the Conference on Advances in Neural Information Processing Systems II. 1999, 368–374

22. Li Y F, Kwok J T, Zhou Z H. Semi-supervised learning using label mean. In: Proceedings of the 26th International Conference on Machine Learning. 2009, 633–640

23. Blum A, Chawla S. Learning from labeled and unlabeled data using graph mincuts. In: Proceedings of the 18th International Conference on Machine Learning. 2001, 19–26

24. Chapelle O, Weston J, Schölkopf B. Cluster kernels for semi-supervised learning. In: Proceedings of the 15th International Conference on Neural Information Processing Systems. 2003, 601–608

25. Szummer M, Jaakkola T. Partially labeled classification with Markov random walks. In: Proceedings of the 14th International Conference on Neural Information Processing Systems. 2002, 945–952

26. Kemp C, Griffiths T L, Stromsten S, Tenenbaum J B. Semi-supervised learning with trees. In: Proceedings of the 16th International Conference on Neural Information Processing Systems. 2004, 257–264

27. Wang H, Wang S B, Li Y F. Instance selection method for improving graph-based semi-supervised learning. In: Proceedings of the 14th Pacific Rim International Conference on Artificial Intelligence. 2016, 565–573

28. Jebara T, Wang J, Chang S F. Graph construction and b-matching for semi-supervised learning. In: Proceedings of the 26th International Conference on Machine Learning. 2009, 441–448

29. Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Proceedings of the 14th International Conference on Neural Information Processing Systems. 2002, 585–591

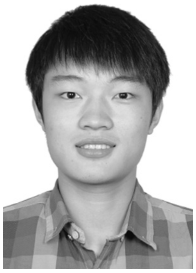
30. Kuncheva L I, Whitaker C J, Shipp C A, Duin R P. Limits on the majority vote accuracy in classifier fusion. Pattern Analysis and Applications, 2003, 6(1): 22–31

31. Delalleau O, Bengio Y, Roux N L. Efficient non-parametric function induction in semi-supervised learning. In: Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics. 2005, 96–103

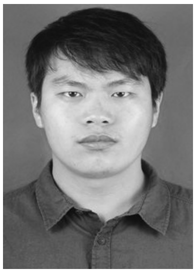
32. Li Y F, Zhou Z H. Improving semi-supervised support vector machines through unlabeled instances selection. In: Proceedings of the 25th AAAI Conference on Artificial Intelligence. 2011, 386–391

33. Yang Y, Nie F P, Xu D, Luo J B. Zhuang Y T, Pan Y H. A multimedia retrieval framework based on semi-supervised ranking and relevance feedback. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(4): 723–742

34. Yang Y, Ma Z G, Nie F P, Chang X J, Hauptmann A G. Multi-class active learning by uncertainty sampling with diversity maximization. International Journal of Computer Vision, 2015, 113(2): 113–127



Hai Wang is a master student at Department of Computer Science and Technology in Nanjing University, China. He is currently a member of the LAMDA Group. His main research interest is machine learning.



Shao-Bo Wang is a master student at Department of Computer Science and Technology in Nanjing University, China. He is currently a member of the LAMDA Group. His main research interest is machine learning.

Yu-Feng Li is currently an associate researcher at Department of Computer Science and Technology in Nanjing University, China. He is currently a member of the LAMDA Group. His main research interests include machine learning and data mining. He won the Microsoft Fellowship Award in 2009 and the Excellent Doctoral Dissertation Award of Chinese Computer Federation in 2013. He has been a senior program committee member of several conferences including IJCAI'17 and IJCAI'15, and served as program committee member for ICML'16, KDD'16, CVPR'16, etc.