Instance Selection Method for Improving Graph-based Semi-Supervised Learning[*]

Hai Wang, Shao-Bo Wang, and Yu-Feng Li

National Key Laboratory for Novel Software Technology, Nanjing University,
Nanjing, 210023, China
{wanghai,wangsb,liyf}@lamda.nju.edu.cn

Abstract. Graph-based semi-supervised learning (GSSL) is one of the most important semi-supervised learning (SSL) paradigms. Though GSSL methods are helpful in many situations, they may hurt performance when using unlabeled data. In this paper, we propose a new GSSL 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 to exploit all the unlabeled instances; instead, we should exploit the unlabeled instances which are highly possible to help improve the performance, while do not take the ones with high risk into account. Experiments on a board range of data sets show that the chance of performance degeneration of our proposal is much smaller than that of many state-of-the-art GSSL methods.

Keywords: graph-based semi-supervised learning · performance degeneration · instance selection

1 Introduction

In many applications, there are plentiful unlabeled training data while the acquisition of class labels is costly and difficult. For example, in webpage categorization (Zhou et al. 2004), manually labeled webpages are always a very small part of the entire web, and unlabeled webpages are in a large part. SSL (Zhu 2007; Chapelle et al. 2006) 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, SSL has attracted significant attentions in machine learning community. One evidence is that three representative works in SSL (Blum and Mitchell 1998; Joachims 1999; Zhu et al. 2003) have won the 10-Year Best Paper Award by ICML in 2008, 2009 and 2013, respectively.

Among many SSL approaches, GSSL is one of the most important SSL paradigms. This line of methods is generally based upon an assumption that similar instances should be shared by similar labels. It encodes both the labeled

[*] This research was supported by NSFC (61403186), JiangsuSF (BK20140613), 863 Program (2015AA015406).

and unlabeled instances as vertices in a weighted graph, with edge weights encoding the similarity between instances. GSSL method aims to assign the labels to unlabeled instances such that the inconsistency with respect to the graph is minimized.

Previous studies generally expected that when the amount of labeled data is limited, GSSL (Zhou et al. 2004; Zhu et al. 2003; Joachims 2003; Blum and Chawla 2001; Camps-Valls et al. 2007; Zhu et al. 2005) could be an effective approach to improve the performance by exploiting auxiliary unlabeled data. However, in many cases (Zhou et al. 2004; Belkin and Niyogi 2004; Karlen et al. 2008; Wang and Zhang 2008; Li et al. 2016), GSSL algorithms using auxiliary unlabeled data might even decrease the learning performance. To enable GSSL to be accepted by more users in more application areas, it is desirable to reduce the chances of performance degeneration when using unlabeled data.

In this paper, we propose an instance selection method in order to reduce the chances of performance degeneration when using unlabeled data. Our basic idea is that given a set of unlabeled instances, it is not the best to exploit all the unlabeled instances; instead, we should exploit the unlabeled instances which are highly possible to help improve the performance, while do not take the ones with high risk into account. We propose our GsslIs (Graph Semi-Supervised Learning with Instance Selection) method which exploits both the predictive label and confidence simultaneously. Experiments on a board range of data sets show that the chance of performance degeneration of our proposal is much smaller than that of many state-of-the-art GSSL methods.

We organize the paper as follows. Section 2 briefly introduces the background. Section 3 presents our method. Experimental results are reported in Section 4. Finally, Section 5 concludes this paper.

2 Background

For the simplicity of notations, let $D = \{\{\boldsymbol{x}_i, y_i\}_{i=1}^{l}, \{\boldsymbol{x}_j\}_{j=l+1}^{l+u}\}$ denote the training data set 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. $y_i \in \{+1, -1\}$ corresponding the label of instance $\boldsymbol{x}_i$, $i = 1, \ldots, l$. In GSSL, 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 training instances. In this following, we briefly introduce two classical GSSL methods. One is the Class Mass Normalization (CMN) method (Zhu et al. 2003) and the other is the Learning with Local and Global Consistency (LLGC) method (Zhou et al. 2004).

CMN defines a function $f : L \cup U \to \mathcal{R}$ over the nodes. According to the intuition of GSSL, similar instances have similar labels and this motivates the choice of the quadratic energy function

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

where $\boldsymbol{\Delta}$ is the Laplacian matrix of graph $G(V, \boldsymbol{W})$ (Belkin and Niyogi 2002). To minimize Equation 1, since it is a convex quadratic form, its optimal solution can be formulated as a closed-form $\boldsymbol{f}_U = (-\boldsymbol{\Delta}_{UU})^{-1}\boldsymbol{\Delta}_{UL}\boldsymbol{y}_L$, the two matrices ($\boldsymbol{\Delta}_{UU}$ and $\boldsymbol{\Delta}_{UL}$) are partitioned from $\boldsymbol{\Delta}$.

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 results. Besides, rather than the CMN method which enforces that the prediction of GSSL 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.

3 Our Proposed Method

Classical GSSL studies (such as, the CMN method and the LLGC method) generally expected that when the amount of labeled data is limited, GSSL could improve the performance by exploiting auxiliary unlabeled data. However, in many empirical cases (Zhou et al. 2004; Belkin and Niyogi 2004; Karlen et al. 2008; Wang and Zhang 2008; Li et al. 2016), GSSL algorithms using auxiliary unlabeled data might even decrease the learning performance. To enable GSSL to be accepted by more users, it is desired to reduce the chances of performance degeneration when using unlabeled data in GSSL.

To address this problem, our basic idea is that given a set of unlabeled instances, it is not the best to exploit all the unlabeled instances without any sanity check; instead, we should exploit the unlabeled instances which are highly possible to help improve the performance, while do not take the ones with high risk into account. Based on this recognition, in the following, we first present two direct approaches based on predictive label aggregation and predictive confidence aggregation respectively, to reduce the chances of performance degeneration. Then, by examining the limitations of these two direct approaches, we propose GsslIs method with the use of both the predictive label and confidence simultaneously.

3.1 Two Direct Approaches

MV The first direct approach is the use of MV (Majority Voting) strategy (Kuncheva et al. 2003) which is known as an effective approach to improve the robustness of a learning method. It aggregates multiple predictive labels from multiple GSSL methods (for example, by using multiple graphs). The label of unlabeled instance is assigned to the majority one among multiple predictive labels.

DirA DirA (Direct Aggregation) is motivated by predictive confidence aggregation, where the confidence obtained by GSSL method can be regarded as a measurement of the reliability of unlabeled data. Formally, let $\boldsymbol{f}_m$ 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 to use and the ones with a low confidence value (or a low rank) are risky and not exploited.

3.2 The GsslIs Method

For the MV method, it only considers the hard label aggregation and may be risky when some hard labels are with low confidences. For the DirA method, it only considers the mean of the predictive values whereas ignores their variance, which might be misled and risky. To alleviate the above deficiencies, we propose the GsslIs method. Our basic observation is that, the MV and the DirA methods are complementary to each other. Specifically, the predictive value aggregation used in the DirA method is able to avoid low confident 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. Based on this observation, the proposed GsslIs method is quite simple and easy to implement. As Algorithm 1 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.

Algorithm 1 The Proposed GsslIs Method

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

Table 1. Experimental Data Sets

Data	#Dim	#Pos	#Neg	#Total	Data	#Dim	#Pos	#Neg	#Total
text	11960	750	750	1500	liverDisorders	6	200	145	345
credit-approval	15	383	307	690	spambase	57	1813	2788	4601
hill-valley	100	606	606	1212	vehicle	16	218	217	435
breastw	9	239	444	683	statlog-heart	13	120	150	270
house-votes	16	267	168	435	house	16	108	124	232
digit1	241	734	766	1500	german	24	300	700	1000
wdbc	14	357	212	569	diabetes	8	500	268	768
isolet	51	300	300	600	horse-colic	25	136	232	368

4 Experiments

4.1 Data Sets

To verify the effectiveness of our proposed method, we evaluate the GsslIs method on a broad range of data sets[1] (Table 1). For each data set, 10 examples are randomly chosen as the labeled examples, and use the remaining data as unlabeled data. The experiments are repeated for 30 times and the average accuracies with their standard deviations are recorded.

4.2 Compared Method

The proposed method is compared with the following methods.

- 1NN: The supervised 1 Nearest Neighbor method, which is used as a baseline supervised approach in classical GSSL (Zhou et al. 2004).
- LLGC: The Learning with Local and Global Consistency method (Zhou et al. 2004).
- CMN: The Class Mass Normalization method (Zhu et al. 2003).
- MV: The majority voting method mentioned in Section 3.1.
- DirA: The direct aggregation method mentioned in Section 3.1.

For 1NN method, Euclidean distance metric is used to locate the nearest neighbors. For CMN and LLGC method, 5 nearest neighbor graphs under 3 kinds of distance metrics (namely Euclidean distance, Cosine distance, Manhattan distance) are conducted for comparison. The parameter of the CMN method is set to the recommended one in the package[2]. The LLGC method is implemented by ourself 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 used in CMN and LLGC method are employed as the set of graphs. In our proposed GsslIs method, the parameter λ is set to 0.7 for all the experimental cases.

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

[2] `http://pages.cs.wisc.edu/~jerryzhu/pub/harmonic_function.m`

Table 2. Accuracy (mean ± std) on 10 labeled examples based on CMN method. For the GSSL methods if the performance is significantly better/worse than 1NN, the corresponding entries are bolded/boxed (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.

Data	1NN		CMN		MV	DirA	GsslIs
	Euclidean	Euclidean	Cosine	Manhattan			
text	59.5±3.4	**64.0±4.8**	**63.1±4.6**	$\boxed{51.2\pm3.2}$	**64.2±4.9**	**62.9±4.2**	**64.5±4.5**
credit	72.9±6.9	$\boxed{69.9\pm8.1}$	$\boxed{68.2\pm7.1}$	$\boxed{69.1\pm8.5}$	$\boxed{69.6\pm7.7}$	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**	$\boxed{73.1\pm4.4}$	**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	$\boxed{77.3\pm5.2}$	$\boxed{73.8\pm4.3}$	$\boxed{78.6\pm5.3}$	**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	$\boxed{61.5\pm1.6}$	$\boxed{61.7\pm1.2}$	$\boxed{61.6\pm1.1}$	$\boxed{61.4\pm1.2}$	**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	$\boxed{88.0\pm2.1}$	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**	$\boxed{62.5\pm4.5}$	**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**

Table 3. Accuracy (mean ± std) on 10 labeled examples based on LLGC method.

Data	1NN		LLGC		MV	DirA	GsslIs
	Euclidean	Euclidean	Cosine	Manhattan			
text	59.5±3.4	$\boxed{55.8\pm4.8}$	57.6±5.9	$\boxed{50.7\pm1.2}$	$\boxed{56.2\pm5.4}$	**62.9±4.4**	**61.0±4.9**
credit	72.9±6.9	$\boxed{69.9\pm7.6}$	$\boxed{68.8\pm7.7}$	$\boxed{69.6\pm8.7}$	$\boxed{68.8\pm7.8}$	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**	$\boxed{78.2\pm7.0}$	**95.9±0.6**	**95.6±0.5**	93.3±3.4	**93.6±2.9**
house-v	86.7±3.0	$\boxed{82.9\pm8.1}$	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	$\boxed{71.0\pm6.8}$	$\boxed{70.9\pm6.1}$	$\boxed{67.1\pm3.7}$	$\boxed{69.2\pm4.9}$	**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	$\boxed{65.6\pm4.7}$	66.3±4.6	$\boxed{64.5\pm3.3}$	$\boxed{65.0\pm4.0}$	**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	$\boxed{60.6\pm5.7}$	$\boxed{59.2\pm4.7}$	$\boxed{59.2\pm5.3}$	$\boxed{59.7\pm5.7}$	**75.3±4.4**	**75.1±5.2**
house	89.4±2.1	$\boxed{80.8\pm9.7}$	$\boxed{83.3\pm8.4}$	$\boxed{79.9\pm9.1}$	$\boxed{82.4\pm8.3}$	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**	$\boxed{62.6\pm4.4}$	**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	$\boxed{63.0\pm3.4}$	$\boxed{63.5\pm4.0}$	$\boxed{62.6\pm3.1}$	$\boxed{62.4\pm2.4}$	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**

4.3 Comparison Results

Table 2 shows the comparison results based on the implementation of CMN. As can be seen, GsslIs achieves highly competitive performance with compared methods. For example, in terms of average accuracy, GsslIs obtains the best average accuracy. While more importantly, the compared GSSL methods all will significantly decrease the performance in many cases, while our proposed approach never degenerate the performance. Both the MV and the DirA method are capable of reducing the chances of performance degeneration, however, they still degenerate the performance in multiple cases, while our proposed method does not have such kind of phenomena. As for 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 compared GSSL methods.

Overall, these results show that our proposed method is able to reduce the chances of performance degeneration, while still obtains highly competitive performance improvement as state-of-the-art GSSL methods.

4.4 Influence on the Number of Graphs

We further study the influence on the number of candidate graphs. We generate the candidate graphs as followings. For each instance, the number of nearest neighbors is randomly picked up from 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. Figure 1 shows the results with different number of candidate graphs. As can be seen, our GsslIs method rarely hurts the performance as the number of candidate graphs varies.

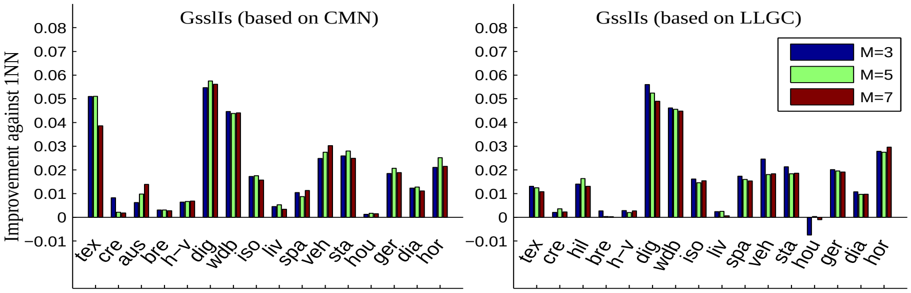

Fig. 1. Influence on the number of graphs M on the performance improvement of GsslIs against 1NN method.

5 Conclusion

In this paper, we propose a new GSSL method GsslIs based on instance selection in order to reduce the chances of performance degeneration for GSSL. Our basic idea is that given a set of unlabeled instances, it is not the best to exploit all the unlabeled instances; instead, we should exploit the unlabeled instances which are highly possible to help improve the performance, while do not take the ones with high risk into account. Experiments on a board range of data sets show that the chance of performance degeneration of our proposal is much smaller than that of some classical GSSL method, while maintains similar performance improvement as many state-of-the-art GSSL approaches. In future we will study the mathematical foundation of GsslIs and extend it to multi-class scenario.

References

Belkin, M., Niyogi, P.: Semi-Supervised Learning on Riemannian Manifolds. Machine Learning, 56, 209-239 (2004)

Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. Advances in Neural Information Processing Systems 14, pp. 585-591. MIT Press (2002)

Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the 7th Annual Conference on Computational Learning Theory. Madison, WI (1998)

Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph mincuts. In: Proceedings of the 18th International Conference on Machine Learning. pp. 19-26. Williamstown, MA (2001)

Camps-Valls, G., Marsheva, T.V.B., Zhou, D.: Semi-supervised graph-based hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 45(10), 3044-3054 (2007)

Chapelle, O., Schölkopf, B., Zien, A., et al.: Semi-supervised learning. (2006)

Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of the 16th International Conference on Machine Learning. pp. 200-209. Bled, Slovenia (1999)

Joachims, T.: Transductive learning via spectral graph partitioning. In: Proceedings of the 20th International Conference on Machine Learning. pp. 290-297. Washington, DC (2003)

Karlen, M., Weston, J., Erkan, A., Collobert, R.: Large Scale Manifold Transduction. In: Proceedings of the 25th International Conference on Machine Learning. pp. 775-782. Helsinki, Finland (2008)

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

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, New York, NY (2016)

Wang, F., Zhang, C.: Label propagation through linear neighborhoods. IEEE Transactions on Knowledge and Data Engineering, 20(1), 55-67 (2008)

Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. Advances in Neural Information Processing Systems 16, pp. 595-602. MIT Press, Cambridge, MA (2004)

Zhu, X.: Semi-supervised learning literature survey. Tech. rep., University of Wisconsin-Madison (2007)

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. pp. 912–919. Washington, DC (2003)

Zhu, X., Lafferty, J., Rosenfeld, R.: Semi-supervised learning with graphs. Ph.D. thesis, Carnegie Mellon University (2005)