Semi-supervised Learning with Ensemble Learning and Graph Sharpening

Inae Choi and Hyunjung Shin*

Department of Industrial & Information Systems Engineering, Ajou University
5 Wonchun-dong, Yeongtong-gu, Suwon, 443-749, Korea
{inae21,shin}@ajou.ac.kr

Abstract. The generalization ability of a machine learning algorithm varies on the specified values to the model-hyperparameters and the degree of noise in the learning dataset. If the dataset has a sufficient amount of labeled data points, the optimal value for the hyperparameter can be found via validation by using a subset of the given dataset. However, for semi-supervised learning--one of the most recent learning algorithms--this is not as available as in conventional supervised learning. In semi-supervised learning, it is assumed that the dataset is given with only a few labeled data points. Therefore, holding out some of labeled data points for validation is not easy. The lack of labeled data points, furthermore, makes it difficult to estimate the degree of noise in the dataset. To circumvent the addressed difficulties, we propose to employ ensemble learning and graph sharpening. The former replaces the hyperparameter selection procedure to an ensemble network of the committee members trained with various values of hyperparameter. The latter, on the other hand, improves the performance of algorithms by removing unhelpful information flow by noise. The experimental results present that the proposed method can improve performance on a publicly available bench-marking problems.

Keywords: Semi-supervised learning, Graph sharpening, Ensemble learning, Hyperparameter selection, Noise reduction.

1 Introduction

In supervised learning, the performance of a model would be improved if the data with class labels were more available, since the model would have more to learn. However, it is often difficult, expensive, and time-consuming to collect the data with labels while unlabeled data is readily available or relatively easy to collect such as in text categorization and protein function classification, etc. One may assume that those unlabeled data also give valuable information for learning. Recently, semi-supervised learning has been proposed to make use of unlabeled data as well as labeled ones by assuming that data with similar attributes lead to similar labels. Originally, this learning framework is to deal with situations where labeled data is only a few while unlabeled data is given in a large quantity. Previous researches have shown that learning with both unlabeled and labeled data can outperform learning with only labeled ones [1][2][3].

C. Fyfe et al. (Eds.): IDEAL 2008, LNCS 5326, pp. 172–179, 2008.

The generalization ability of a model, regardless of supervised learning or semi-supervised learning, varies on the specified values to model-hyperparameter and the degree of noise in the learning dataset. The hyperparameter selection depends on the degree of noise, and so the two problems have been often dealt with together as a single issue. However, either one can solely exist as a separate problem since, for instance, the hyperparameter selection will still remain even after the noisy data points are removed from the dataset. The hyperparameter selection is rather directly related to the complexity of problem in hand which is not likely to be identified in advance. If the dataset has a sufficient amount of labeled data, the optimal value for the hyperparameter can be found in a trial-and-error fashion by checking the validation performance. In supervised learning, cross-validation is generally used for this. However, for semi-supervised learning, it is hardly able to hold out some of labeled data in order to make a separate validation set. Meanwhile, noise in the dataset also increases the problem complexity. A higher degree of noise leads to a more complicated problem and hence a higher model complexity. If the degree of noise is known in advance, overfitting to noise can be prevented by imposing a more penalty on a more complicated model. In supervised learning, even if the degree of noise is not known a priori, the estimation for it is still available by checking the class impurity with labeled data. In semi-supervised learning, however, the noise estimation does not seem to be possible: again, because of lack of labeled data.

In this paper, we propose to employ ensemble learning and graph sharpening to circumvent the addressed difficulties. Ensemble learning is to combine a variety of models so as to improve performance by reducing the bias or variance of error [4][5][6][7]. And graph sharpening is a most recently proposed method in semi-supervised learning, which eliminates or reduces the noisy or corrupt information in the dataset by taking into explicit account the values of relationship between data points [8]. The former replaces the hyperparameter selection procedure to an ensemble network of the committee members trained with various values of hyperparameter. The latter, on the other hand, improves the performance of algorithms by removing or alleviating influence of noise in the dataset.

The paper is organized as follows. In Section 2, we present the basic idea of the proposed method with brief introduction to graph-based semi-supervised learning, graph sharpening, and ensemble learning. In Section 3, we show the results of experiments on synthetic and real-world datasets. We conclude with additional remarks in Section 4.

2 Method

The proposed method is based on graph-based semi-supervised learning. Within this framework, we employ graph sharpening and ensemble learning: First, multiple graphs are generated with various values of hyperparameter. The individual graphs are "sharpened" for de-noising. And then, those graphs are combined into an ensemble network. The following three subsections introduce the methods in due order.

2.1 Graph-Based Semi-supervised Learning

In graph-based semi-supervised learning [9], a data point $x_i\,(i=1,\cdots,n)$ is represented as a node *i* in a graph, and the relationship between data points is represented by an edge (see Fig.1). The connection strength from each node *j* to each other node *i* is encoded in element w_{ij} of a weight matrix *W*. Often, a Gaussian function between points is used to specify connection strength:

$$w_{ij} = \begin{cases} \exp(-\dfrac{(x_i - x_j)^T (x_i - x_j)}{\sigma^2}) & if \quad i \sim j, \\ 0 & otherwise. \end{cases} \tag{1}$$

The *i~j* stands for node *i* and *j* having an edge between them that can be established by *k* nearest neighbors (*kNN*) where *k* is a user-specified hyperparameter. The labeled nodes have labels $y_l \in \{-1,1\}$, while the unlabeled nodes have zeros $y_u = 0$. Our algorithm will output an *n*-dimensional real-valued vector $f = \left[f_l^T f_u^T\right]^T = (f_1,\cdots,f_l,f_{l+1},\cdots,f_{n=l+u})^T$, which can be thresholded to make label predictions on $f_{l=1},\cdots,f_n$ after learning. It is assumed that f_i should be closed to the given label y_i in labeled nodes (loss condition), and overall, f_i should not be too different from the f_j of adjacent nodes (smoothness condition). One can obtain *f* by minimizing the following quadratic function [9][10][11].

$$\min_f (f-y)^T (f-y) + \mu f^T L f \tag{2}$$

where $y = (y_1,\cdots,y_l,0,\cdots,0)^{\mathrm{T}}$, and the matrix *L*, called the *graph Laplacian matrix*, is defined as *L=D-W* where $D = diag(d_i)$, $d_i = \sum_j w_{ij}$. The parameter μ trades off loss versus smoothness. The solution of this problem is obtained as

$$f = (I + \mu L)^{-1} y \,. \tag{3}$$

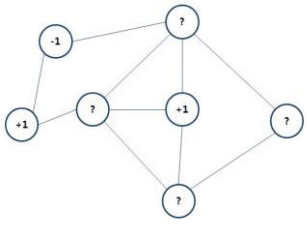


Fig. 1. An ordinary graph by *W*: The labeled node is denoted as "+1" or "-1", and the unlabeled node as "?". The edge has no directionality.

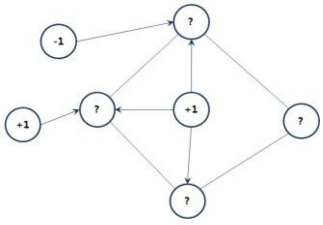


Fig. 2. A sharpened graph by W_s **:** By graph sharpening some edges have been removed or have assumed directionality according to the importance of the information flow

2.2 Graph Sharpening

The recently proposed graph sharpening is an effective method to improve the performance of the graph-based semi-supervised learning algorithms [8]. As described in the earlier section, the relationship between the data points is represented by the similarity matrix W which plays a critical role in prediction as a form of graph-Laplacian L. Related to the matrix, graph sharpening addresses two points. First, the data in many kinds of noise form an unnecessary edge and cause the decline of the performance of the algorithm. Second, the matrix W is dealt with as fixed and symmetric, which means the edge is considered without direction, and the reflected similarity is an undirected edge. When weight matrix W, however, describes the relationships between the labeled and unlabeled points, it is not necessarily desirable to regard all such relationships as symmetric. That is, the contribution of all edges to the information flow may be varied by not weighing them equally. Graph sharpening improves the performance of algorithms by changing the weight matrix to remove the edge caused by noise and to employ directionality between edges by asymmetrically weighing edge-weights [3]. If an ordinary weight matrix is represented as a block matrix $W = \begin{bmatrix} W_{ll} W_{lu} | W_{ul} W_{uu} \end{bmatrix}$, graph sharpening changes the matrix as $W = \begin{bmatrix} diagonal & 0 & | W_{ul} & W_{uu} \end{bmatrix}$ in the simplest case. W_{lu} should be read as the weight of an edge from an unlabeled to a labeled point ($u \rightarrow l$). The output prediction for unlabeled data points can be obtained using the following equation:

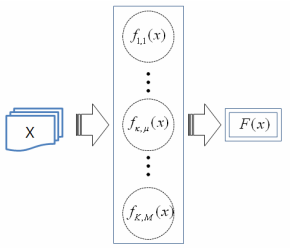
$$f_u = \mu(I + \mu(D_{uu} - W_{uu}))^{-1} W_{ul} y_l \,. \tag{4}$$

Fig. 2 shows change in a graph after sharpening. Note that some edges have been removed and have assumed directionality. A detailed mathematical foundation of graph sharpening can be found in [8].

2.3 Ensemble Learning

Ensemble learning is to combine a variety of member models in order to reduce the bias or variance of error, and to improve performance therefrom [4][5][6][7]. The ensemble network achieves the better performance when its members become more diverse. The members can be diversified by using perturbed learning datasets as in bagging [4] and boosting [12], or by directly perturbing the hyperparameter values of the learning algorithm while keeping a single learning dataset as common across the members [13]. In our method, the latter is taken- diversification by hyperparameter perturbation but no variation in the learning dataset. This becomes of great benefit to semi-supervised learning, particularly with respect to its hyperparameter selection procedure. Using a validation set for hyperparameter selection, the most representative approach which has originally been designed for supervised learning, does not well fit into semi-supervised learning. Because, in semi-supervised learning, the amount of labeled data in the entire dataset is absolutely deficient even for learning, in other words, even for making a training dataset, and so further splitting them for making a validation dataset is hard to be taken as a reasonable approach. In the proposed method, multiple networks are trained with various values of hyperparameter without

using a validation set, and then combined into an ensemble network: we train as many individual networks as all the possible combinations of the two hyperparameters, the number of neighbors $\kappa(\kappa=1,\cdots,K)$ in Eq.(1) and the loss-smoothness tradeoff $\mu(\mu=1,\cdots,M)$ in Eq.(2), and then the final output of the ensemble network is calculated by taking the simple mean of the output values of the $|K|\times|M|$ member networks. This replaces selecting a single best network via a validation set, which becomes a more practical approach for semi-supervised learning. Fig 3 illustrates the procedure.



$$F(x)=\frac{\sum_{\kappa=1}^{K}\sum_{\mu=1}^{M} f_{\kappa,\mu}(x)}{|K|\times|M|}$$
$$\kappa=1,\cdots,K\ \ \mu=1,\cdots,M$$

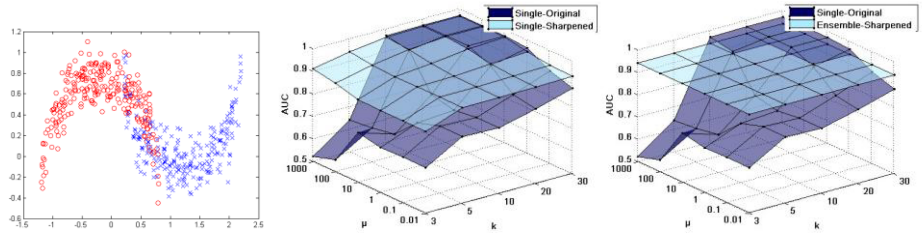
Fig. 3. Ensemble learning

3 Experiment Results

We applied the proposed method, ensemble learning on sharpened graphs, to various kinds of data sets: an artificial dataset and five benchmarking datasets. We examined the change in the area under the ROC curve (AUC) in terms of 'original' versus 'sharpened' and 'single' versus 'ensemble.' This setting enabled us to see the effect of the proposed method from two separate viewpoints, graph sharpening and ensemble learning.

3.1 Artificial Data

The proposed method was evaluated on the two-moon toy problem as shown in Fig. 4(a). A total of 500 input data were generated from two classes, each with 245 unlabeled and 5 labeled data. The AUC was measured under various combinations of hyperparameters such as (k, i) $\in$ {3, 5, 10, 20, 30} × {0.01, 0.1, 1.0, 10, 100, 1000}, where k and i indicate the number of k-nearest neighbors in Eq.(1) and the loss-smoothness tradeoff parameter in Eq.(2), respectively. Fig. 4(b) and (c) depict the changes in the AUC over the hyperparameter variation. Fig.4(b) shows the effect of graph sharpening: when compared with single-original, single-sharpened is less sensitive to hyperparameter variation because of noise reduction by graph sharpening. Also note that in every comparison the AUC is increased after the original graph is sharpened. Fig.4(c) shows that the synergy effect of graph sharpening and ensemble learning: when compared with single-original, ensemble-sharpened shows less sensitivity and higher accuracy. Also, when compared with single-sharpened, ensemble-sharpened gives a more stabilized performance.



(a) (b) (c)

Fig. 4. Artificial data: (a) presents the two-moon toy problem. (b) and (c) depict the changes in the AUC over hyperparameter variation (k and μ), "(b) single-original vs. single-sharpened" and "(c) single-original vs. ensemble-sharpened", respectively.

3.2 Real Data

Table 1 shows the AUC comparison results between "single-original" and "ensemble-sharpened" for five real-world datasets [15]. Each dataset has two sets of predetermined 12 splits, one is for 10 labeled data and the other is for 100 labeled data. The AUC was measured at every combination of hyperparameters (k, i) $\in$ {3, 5, 10, 20, 30} × {0.01, 0.1, 1.0, 10, 100, 1000}. The Wilcoxon signed-rank test was used to verify the performances of both methods, where a smaller the value of p stands for a more significant difference between them [14]. The values listed in the table are the mean and standard deviation of AUC values across the 12 splits in datasets. Most in the cases, the proposed method increased AUCs and the effect was statistically significant. The maximum avg. increase in AUC, 0.10, was obtained from USPS-100 labeled dataset. On the other hand, the minimum avg. increase was 0 from BCI-10 labeled dataset guaranteeing 'no loss' even in the worst case. The five pairs of bar graphs in Fig.5 visualize the results.

Table 1. AUC comparison for the five benchmark data sets (mean ± std)

Dataset (dimension, number of points)		Single-Original	Ensemble-Sharpened (proposed method)	p-value
(1)Digit1 (241, 1500)	10label	0.89 ± 0.04	0.93 ± 0.05	0.00
	100label	0.97 ± 0.02	0.99 ± 0.01	0.00
(2)USPS (241, 1500)	10label	0.65 ± 0.09	0.68 ± 0.10	0.00
	100label	0.87 ± 0.08	0.97 ± 0.01	0.00
(3)BCI (117,400)	10label	0.50 ± 0.01	0.50 ± 0.03	0.93
	100label	0.53 ± 0.03	0.56 ± 0.02	0.00
(4)g241c (241, 1500)	10label	0.55 ± 0.03	0.56 ± 0.05	0.00
	100label	0.63 ± 0.05	0.65 ± 0.04	0.00
(5)g241n (241, 1500)	10label	0.55 ± 0.03	0.56 ± 0.04	0.00
	100label	0.63 ± 0.04	0.65 ± 0.04	0.00

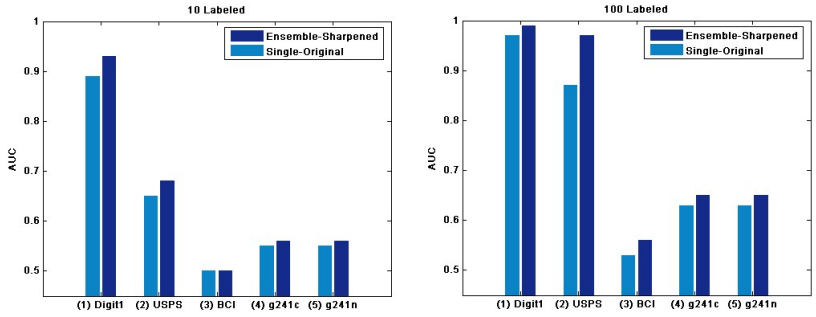


Fig. 5. The AUC comparison of single-original and ensemble-sharpened

4 Conclusion

In this paper, we proposed to use ensemble learning and graph sharpening for graph-based semi-supervised learning. Instead of using a single graph specified to a certain value of the hyperparameter in a trial-and-error fashion, we employed a graph ensemble of which committee members trained with various values of the hyperparameter. Ensemble learning stabilizes performance by reducing the error variance-and-bias of individual learners. Additionally, with ensemble learning, the hyperparameter selection procedure becomes less critical. The accuracy of an individual graph, on the other hand, was improved by the graph sharpening. Graph sharpening removes noisy or unnecessary edges from the original graph. This enhances the robustness to noise in the dataset. When applied to an artificial problem and five real-world problems, the synergy of ensemble learning and the graph sharpening resulted in more significant improvement in performance.

Acknowledgements

The authors would be like to gratefully acknowledge support from Post Brain Korea 21 and the research grant from Ajou University.

References

1. Zhu, X.: Semi-supervised learning with graphs. Ph.D. dissertation, Carnegie Mellon University (2005)
2. Shin, H., Tsuda, K.: Prediction of Protein Function from Networks. In: Chapelle, O., Schoelkopf, B., Zien, A. (eds.) Book: Semi-Supervised Learning, Ch. 20, pp. 339–352. MIT Press, Cambridge (2006)
3. Shin, H., Lisewski, A.M., Lichtarge, O.: Graph Sharpening plus Graph Integration: A Synergy that Improves Protein Functional Classification. Bioinformatic 23(23), 3217–3224 (2007)
4. Breiman, L.: Bagging Predictors. Machine Learning 24, 123–140 (1996)

5. Perrone, M.P.: Improving Regression Estimation: Averaging Methods for Variance Reduction with Extension to General Convex Measure Optimization. Ph.D Thesis, Brown University, Providence, RI (1993)
6. Sharkey, A.J.C.: Combining Diverse Neural Nets. The Knowledge Engineering Review 12(3), 231–247 (1997)
7. Tumer, K., Ghosh, J.: Error Correlation and Error Reduction in Ensemble Classifiers. Connection Science 8(3), 385–404 (1996)
8. Shin, H., Hill, N.J., Raetsch, G.: Graph-based semi-supervised learning with sharper edges. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 402–413. Springer, Heidelberg (2006)
9. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. Advances in Neural Information Processing Systems (NIPS) 16, 321–328 (2004)
10. Belkin, M., Matveeva, I., Niyogi, P.: Regularization and regression on large graphs. In: Shawe-Taylor, J., Singer, Y. (eds.) COLT 2004. LNCS (LNAI), vol. 3120, pp. 624–638. Springer, Heidelberg (2004)
11. Chapelle, O., Weston, J., Schölkopf, B.: Cluster kernels for semi-supervised learning. In: Advances in Neural Information Processing Systems (NIPS), vol. 15, pp. 585–592 (2003)
12. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Machine Learning: Proceedings of the Thirteenth International Conference, pp. 148–156 (1996)
13. Shin, H., Cho, S.: Pattern selection using the bias and variance of ensemble. Journal of the Korean Institute of Industrial Engineers 28(1), 112–127 (2001)
14. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
15. http://www.kyb.tuebingen.mpg.de/ssl-book/benchmarks.html