ACTIVE MODEL SELECTION FOR GRAPH-BASED SEMI-SUPERVISED LEARNING

Bin Zhao, Fei Wang, Changshui Zhang, Yangqiu Song

State Key Laboratory of Intelligent Technologies and Systems,
Tsinghua National Laboratory for Information Science and Technology (TNList),
Department of Automation,Tsinghua University, Beijing 100084, China

ABSTRACT

The recent years have witnessed a surge of interest in *Graph-Based Semi-Supervised Learning (GBSSL)*. However, despite its extensive research, there has been little work on graph construction, which is at the heart of *GBSSL*. In this study, we propose a novel active learning method, *Active Model Selection (AMS)*, which aims at learning both data labels and the optimal graph by allowing the learner the flexibility to choose samples for labeling. *AMS* minimizes the regularization function in *GBSSL* by iterating between the *active sample selection* step and the *graph reconstruction* step, where the samples querying which leads to the optimal graph are selected. Experimental results on four real-world datasets are provided to demonstrate the effectiveness of *AMS*.

Index Terms— Graph Based Semi-Supervised Learning (GBSSL), Model Selection, Active Learning, Gaussian Function, Gradient Descent

1. INTRODUCTION

In many practical applications of pattern classification and data mining, one often faces a lack of sufficient labeled data, since labeling often requires expensive human labor and much time. However, in many cases, large number of unlabeled data can be far easier to obtain. For example, in text classification, one may have an easy access to a large database of documents (*e.g.* by crawling the web), but only a small part of them are classified by hand. Consequently, *Semi-Supervised Learning (SSL)* methods, which aim to learn from partially labeled data, are proposed[1].

In recent years, *GBSSL* has become one of the most active research areas in *SSL* community [2]. *GBSSL* uses a graph $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$ to describe the structure of a dataset, where $\mathcal{V}$ is the node set corresponding to the labeled and unlabeled samples, and $\mathcal{E}$ is the edge set. In most of the traditional methods [3, 4, 5], each edge $e_{ij} \in \mathcal{E}$ is associated with a weight w_{ij}, which reflects the similarity between pairwise samples. The weight is usually computed by certain parametric function, *i.e.*,

$$w_{ij} = h_\theta(\mathbf{x}_i, \mathbf{x}_j, \theta) \tag{1}$$

Here, a specific choice of h_θ and related parameters θ is called a model, with which we construct the graph. The choice of the model can affect the final classification result significantly, which can be seen from the toy example shown in Fig. 1, where h_θ is fixed to *Gaussian function*,

$$w_{ij} = \exp\ (-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2)) \tag{2}$$

and classification results with different values of variance σ are shown. However, as pointed out by [1], although at the heart of *GBSSL*, model selection is still a problem that has not been well studied.

To address such a problem, we propose an active learning method, *Active Model Selection (AMS)*, which aims at learning both data labels and the optimal model by allowing the learner the flexibility to choose samples for labeling. Traditionally, active learning methods aim to query samples that could decrease most the generalization error of the resulting classifier. However, since graph construction is at the heart of *GBSSL*, the active learning method we employ here targets to select the most informative samples for model selection. More concretely, the *AMS* algorithm selects samples, querying which could lead to the optimal model. The *AMS* algorithm first establishes an objective function composed of two parts, *i.e.* the *smoothness* and the *fitness* of the data labels, to measure how good the classification result of the *Semi-Supervised Learning* task is. Then *AMS* will minimize this objective function by alternating between the *active sample selection* step and the *graph reconstruction* step. Fig. 2 presents the flow charts of traditional active learning methods and *AMS* to show the difference.

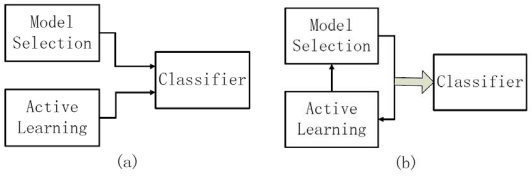

Fig. 2. Flow charts of (a) traditional active learning methods and (b) *AMS*.

The rest of this paper is organized as follows. In section 2, we introduce some works related to this paper. The *AMS* algorithm is presented in detail in section 3. In section 4,

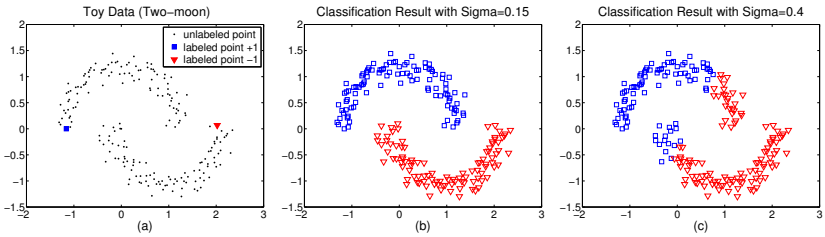

Fig. 1. Classification results on the two-moon pattern using the method in [3], a powerful transductive approach operating on graph with the edge weights computed by a Gaussian function. (a) toy data set with two labeled points; (b) classification results with $\sigma = 0.15$; (c) classification results with $\sigma = 0.4$. We can see that a small variation of σ will cause a dramatically different classification result.

we provide experimental results on four real-world datasets, followed by the conclusions in section 5.

2. NOTATIONS AND RELATED WORKS

In this section we will introduce some notations and briefly review some related works of this paper.

Given a point set $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_l, \mathbf{x}_{l+1}, \cdots, \mathbf{x}_n\}$ and a label set $\mathcal{T} = \{-1, +1\}$ (generalization to multi-class scenario can be obtained in the same manner), where the first l points in $\mathcal{X}$ are labeled as $y_i \in \mathcal{T}$, while the remaining points are unlabeled. Our goal is to predict the labels of the unlabeled points[1]. We denote the initial labels in the dataset by an $n \times 1$ vector $\mathbf{y}$ with $y_i = 1$ or -1 if $\mathbf{x}_i$ is labeled as positive or negative, and 0 if $\mathbf{x}_i$ is unlabeled. The classification result on the dataset $\mathcal{X}$ is also represented as an $n \times 1$ vector $\mathbf{f} = [f_1, \ldots, f_n]^T$, which determines the label of $\mathbf{x}_i$ by $y_i = \mathrm{sgn}(f_i)$. In *GBSSL*, we construct the $n \times n$ *weight matrix* W for graph $\mathcal{G}$ with its (i,j)-th entry $W_{ij} = w_{ij}$ computed by Eq.(1), and $W_{ii} = 0$. The *degree matrix* D for graph $\mathcal{G}$ is defined as an $n \times n$ diagonal matrix with its (i,i)-entry equal to the sum of the i-th row of W. Finally, the *normalized graph Laplacian* [7] for graph $\mathcal{G}$ is defined as $L = I - S = I - D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$.

Based on the above preliminaries, *Zhou et al.* proposed the following regularization function [3] for *GBSSL*:

$$Q = (\mathbf{f} - \mathbf{y})^T(\mathbf{f} - \mathbf{y}) + \lambda \mathbf{f}^T(I - S)\mathbf{f} \tag{3}$$

The first term in Eq.(3) restricts that a good classifying function should not change too much from the initial label assignment, and the second term measures the *smoothness* of the data labels. The regularization parameter $\lambda > 0$ adjusts the tradeoff between these two terms. Thus, the optimal classification function can be obtained as: $\mathbf{f}^* = \arg\min_{\mathbf{f}} Q = (1-\alpha)(I - \alpha S)^{-1}\mathbf{y}$, where $\alpha = \frac{\lambda}{1+\lambda}$. By letting $\mathbf{f} = \mathbf{f}^*$ in Eq.(3), regularization function Q is fully determined by initial labels $\mathbf{y}$ and the model $\{h_\theta, \theta\}$

$$Q(\mathbf{y}, h_\theta, \theta) = \mathbf{y}^T[I - (1-\alpha)(I - \alpha S)^{-1}]\mathbf{y} = \mathbf{y}^T A \mathbf{y} \tag{4}$$

[1]In this paper we concentrate on the transductive setting. One can easily extend our algorithm to inductive setting using the method introduced in [6].

where $A = I - (1-\alpha)(I - \alpha S)^{-1}$ depends on h_θ and θ. As we noted in section 1, one of the problems existing in these graph based methods is that the model (*i.e.* h_θ and θ in Eq.(1)) can affect the final classification results significantly. Moreover, as shown in Eq.(4), the model and the labels $\mathbf{y}$ are dependent. Specifically, the optimal model $\{h_\theta, \theta\}$ relies on the vector $\mathbf{y}$.

3. ACTIVE MODEL SELECTION

In this section, we first propose a gradient descent based model selection method for *GBSSL*. Then we provide details of the *Active Model Selection* algorithm.

3.1. Model Selection via Gradient Descents

We fix the parametric function h_θ to *Gaussian Function* in this paper, as shown in Eq.(2) and select the optimal variance σ. The derivative of Q w.r.t. σ can be calculated as follows

$$\frac{\partial Q(\mathbf{y}, \sigma)}{\partial \sigma} = -\alpha(1-\alpha)\mathbf{y}^T(I - \alpha S)^{-1}\frac{\partial S}{\partial \sigma}(I - \alpha S)^{-1}\mathbf{y} \tag{5}$$

Since $S_{ij} = \frac{W_{ij}}{\sqrt{D_{ii}D_{jj}}}$, we get

$$\frac{\partial S_{ij}}{\partial \sigma} = \frac{\widetilde{W}_{ij}}{\sqrt{D_{ii}D_{jj}}} - \frac{1}{2}\frac{W_{ij}\widetilde{D}_{ii}}{\sqrt{D_{ii}^3 D_{jj}}} - \frac{1}{2}\frac{W_{ij}\widetilde{D}_{jj}}{\sqrt{D_{ii}D_{jj}^3}} \tag{6}$$

$$\widetilde{W}_{ij} \triangleq \frac{\partial W_{ij}}{\partial \sigma} = \frac{\partial \exp(-\frac{d_{ij}^2}{2\sigma^2})}{\partial \sigma} = \frac{d_{ij}^2 \exp(-\frac{d_{ij}^2}{2\sigma^2})}{\sigma^3} \tag{7}$$

$$\widetilde{D}_{ii} \triangleq \frac{\partial D_{ii}}{\partial \sigma} = \frac{\partial \sum_j W_{ij}}{\partial \sigma} = \sum_j \frac{\partial W_{ij}}{\partial \sigma} \tag{8}$$

where d_{ij} is the distance between samples $\mathbf{x}_i$ and $\mathbf{x}_j$, and we employ Euclidean distance in this paper unless further noticed. With the derivative of Q w.r.t. σ calculated above, the *model selection* problem can be tackled with gradient descent.

3.2. Active Model Selection

Since the optimal model is determined via gradient based method, the most informative samples for model selection would be those that maximize the derivative of the objective

function Q w.r.t. the model hyperparameter, in *Gaussian function*, the variance σ. However, since querying those samples that maximize $|\frac{\partial Q}{\partial \sigma}|$ might increase the objective function, we control the acceptance of such a query by introducing an acceptance probability determined by the increase of Q. After the sample is queried, *AMS* retrains the model and constructs the graph in *GBSSL* with this new model. We assume only one sample is added to $\mathbf{y}$ at a time, therefore $\mathbf{y} = \mathbf{y}^0 + y_k \mathbf{e}_k$, where $\mathbf{y}^0$ is the label vector before querying sample $\mathbf{x}_k$, y_k is the actual label for sample $\mathbf{x}_k$ and $\mathbf{e}_k = (0, \ldots, 0, 1, 0, \ldots, 0)$ is the unit vector with only the k-th element equal to 1.

$$\mathbf{y}^* = \arg\min_{\mathbf{y}} Q = \arg\min_{\mathbf{y}} (Q - Q_0) = \arg\min_{\mathbf{y}} \triangle Q \quad (9)$$

where Q_0 stands for the value of the regularization function before querying the selected sample. The decrease of Q after querying sample $\mathbf{x}_k$ is $\triangle Q = 2y_k[A\mathbf{y}^0](k) + A_{kk}$, where $[A\mathbf{y}^0](k)$ denotes the k-th element of the vector $A\mathbf{y}^0$ with A defined the same as in Eq.(4). The algorithm is as follows:

Initialization. Randomly initialize σ. Iterate between the following two steps until convergence;

Active sample selection. Denote the present value of variance by σ^*, actively select sample $\mathbf{x}_k$, querying which maximizes $\left|\frac{\partial Q(\sigma,\mathbf{y})}{\partial \sigma}|_{\sigma=\sigma^*}\right|$ and calculate $\triangle Q = Q(f^{+\mathbf{x}_k}) - Q_0$. If $\triangle Q \leq 0$, accept $\mathbf{x}_k$ as the next sample for querying; else accept $\mathbf{x}_k$ with probability $P(k) = \exp\left(-\frac{\triangle Q \cdot l}{k_B(l-q)}\right)$, where l is the total number of samples to query, q is the number of samples already queried so far, and k_B is Boltzmann's constant [8], which is chosen to be the largest possible decrease of Q while selecting the first sample. If $\mathbf{x}_k$ is not accepted, check the next sample that leads to $\left|\frac{\partial Q(\sigma,\mathbf{y})}{\partial \sigma}|_{\sigma=\sigma^*}\right|$ only less than $\mathbf{x}_k$. Proceeds until one sample is accepted. Query $\mathbf{x}_k$ and set $\mathbf{y} = \mathbf{y}^0 + y_k \mathbf{e}_k$;

Graph reconstruction. Calculate $\sigma^* = \arg\min_\sigma Q(\sigma, \mathbf{y})$ by gradient descent.

Actually, $P(k)$ is the Boltzmann probability [8] with the temperature selected as $T = \frac{l-q}{l}$. In the above algorithm, instead of discarding those samples querying which might increase Q, they can also be incorporated into the label vector with controlled acceptance probability. Consequently, *AMS* obtains the ability to jump out of local minima. With the temperature T decreasing as more samples are queried, the probability for accepting an uphill step also decreases.

Now we present how to select the sample querying which maximizes $|\frac{\partial Q}{\partial \sigma}|$. According to Eq.(5), the derivative of Q with respect to σ can be computed as $\frac{\partial Q(\sigma,\mathbf{y})}{\partial \sigma} = \mathbf{y}^T B(\sigma)\mathbf{y}$, where $B(\sigma) = -\alpha(1-\alpha)[(I-\alpha S)^{-1}\frac{\partial S}{\partial \sigma}(I-\alpha S)^{-1}]$ only depends on σ. Since in each iteration of *AMS*, the graph reconstruction step optimizes the regularization function Q w.r.t. σ, suppose the present label vector is $\mathbf{y} = \mathbf{y}^0$,

$$\frac{\partial Q(\sigma,\mathbf{y})}{\partial \sigma}|_{\sigma=\sigma^*,\mathbf{y}=\mathbf{y}^0} = {\mathbf{y}^0}^T B(\sigma^*)\mathbf{y}^0 = 0 \quad (10)$$

Hence, we only need to compute the increase of $\frac{\partial Q(\sigma,\mathbf{y})}{\partial \sigma}$ w.r.t. the newly labeled sample. Denote the index of the newly labeled sample by k,

$$\begin{aligned}\frac{\partial Q(\sigma,\mathbf{y})}{\partial \sigma}|_{\sigma=\sigma^*} &= \mathbf{y}^T B(\sigma^*)\mathbf{y} \\ &= 2y_k \sum_{j=1}^{m} B_{kj}(\sigma^*) y_j^0 + B_{kk}(\sigma^*) \\ &= 2y_k[B(\sigma^*)\mathbf{y}^0](k) + B_{kk}(\sigma^*) \quad (11)\end{aligned}$$

where $[B(\sigma^*)\mathbf{y}^0](k)$ denotes the k-th element of the vector $B(\sigma^*)\mathbf{y}^0$. Define the gain for labelling the k-th sample as the increase of the derivative of Q w.r.t. σ after querying it, therefore:

$$G(f^{+(\mathbf{x}_k,y_k)}) = \left|2y_k[B(\sigma^*)\mathbf{y}^0](k) + B_{kk}(\sigma^*)\right| \quad (12)$$

Since we don't know what answer y_k we will receive, we assume the answer is approximated with

$$p_{+1}(y_k) \triangleq p(y_k = 1) \approx \frac{1}{1 + e^{-f_k}} \quad (13)$$

where $p_{+1}(y_k)$ denotes the probability of $y_k = 1$. The expected gain after querying node k is therefore:

$$\begin{aligned}G(f^{+\mathbf{x}_k}) &= p_{-1}(y_k)G(f^{+(\mathbf{x}_k,-1)}) + p_{+1}(y_k)G(f^{+(\mathbf{x}_k,+1)}) \\ &\approx \left(1 - \frac{1}{1+e^{-f_k}}\right)|-2[B(\sigma^*)\mathbf{y}^0](k) + B_{kk}(\sigma^*)| \\ &\quad + \frac{1}{1+e^{-f_k}}|2[B(\sigma^*)\mathbf{y}^0](k) + B_{kk}(\sigma^*)| \quad (14)\end{aligned}$$

Hence, the next sample $\mathbf{x}_k$ is selected as $k = \arg\max_{k'} G(f^{+\mathbf{x}_{k'}})$. After the sample $\mathbf{x}_k$ is selected, *AMS* checks if $\mathbf{x}_k$ is accepted, if not, check the next sample that leads to $\left|\frac{\partial Q(\sigma,\mathbf{y})}{\partial \sigma}|_{\sigma=\sigma^*}\right|$ only less than $\mathbf{x}_k$ until one sample is accepted.

4. EXPERIMENTS

We validate the effectiveness of *Active Model Selection* on four real-world datasets, the Breast Cancer and Ionosphere datasets from UCI database[2], USPS[3], and 20-newsgroup[4] datasets. Breast Cancer dataset contains 683 samples, and Ionosphere contains 351 samples. The USPS handwritten digits dataset contains images of $0, \ldots, 9$ as 10 classes. Finally, we choose the topic *rec* which contains *autos*, *motorcycles*, *baseball* and *hockey* from the 20-newsgroup dataset version 20-news-18828. We preprocess the data in the same manner as [3] and obtain 3970 document vectors in a 8014-dimensional space. For document classification, the distance between points x_i and x_j is defined to be $d(x_i, x_j) = 1 - \frac{x_i \cdot x_j}{||x_i|| ||x_j||}$.

[2] http://www.ics.uci.edu/∼mlearn/

[3] http://www.kernel-machines.org/data.html

[4] http://people.csail.mit.edu/jrennie/20Newsgroups/

4.1. Comparison with Other Model Selection Methods for *GBSSL*

In this section, we compare our *active model selection (AMS)* method with three state-of-the-art *model selection* algorithms for *GBSSL*, i.e., *label entropy minimization (MinEnt)* [5], *leave-one-out cross validation (LOO)* [9] and *evidence maximization (LEM)* [10]. Moreover, to validate the novel active learning framework we propose in this paper, we also compare with the method which simply iterates between traditional active learning and gradient descent based model selection. We call this method *Simple Iterative Combination (SIC)*. Since *MinEnt, LOO* and *LEM* only consider the binary classification scenario, we provide experimental results on 6 two-way classification tasks. Test accuracies averaged over 20 random trials are reported. From Fig. 3 we can clearly see the advantage of *active model selection*, i.e., with the same amount of samples labeled, *AMS* achieves the highest classification accuracy. Moreover, the advantage of *AMS* over *SIC* demonstrates the effectiveness of our active learning framework, i.e., for *GBSSL*, *active model selection* is better than combining traditional active learning and model selection. However, as controlled uphill steps are incorporated in *AMS*, it might take more time to converge than *SIC*.

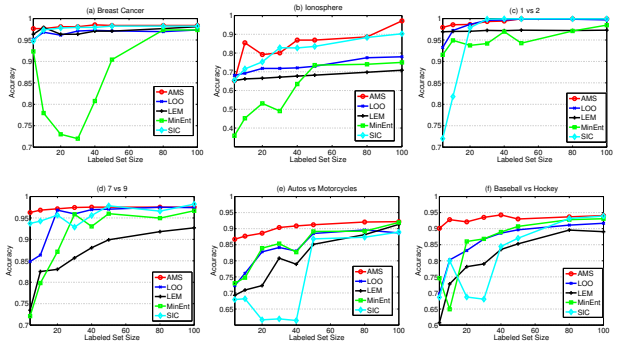

Fig. 3. Test accuracies on UCI, USPS and 20-newsgroup datasets. (a) Breast cancer; (b) Ionosphere; (c) 1 vs 2; (d) 7 vs 9; (e) *autos* vs *motorcycles*; (f) *baseball* vs *hockey*. The number of labeled samples increases from 2 to 100.

4.2. Comparison with Other Classification Methods

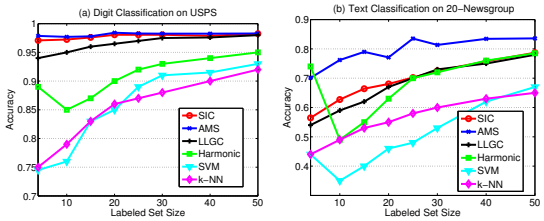

Fig. 4. Multi-category classification accuracies on USPS and 20-newsgroup datasets. (a) Digit recognition with USPS digits dataset for a total of 3874 samples (a subset containing digits from 1 to 4). (b) Text classification with 20-newsgroup dataset for a total of 3970 document vectors.

We compare the performance of *AMS* on multi-category classification tasks with two supervised methods, *k-NN*, *SVM* and two semi-supervised methods, *LLGC* [3] and *harmonic function* [5] in this section. The parameters in *k-NN*, *SVM*, *LLGC* and *harmonic function* are tuned by grid search. The number of labeled samples increases from 4 to 50 and test accuracies averaged over 50 random trials are reported. Figure 4 shows a clear advantage of *AMS* on multi-category classification.

5. CONCLUSIONS

We propose an active learning method, *Active Model Selection*, to solve the model selection problem for *GBSSL*. Different from traditional active learning methods, *AMS* queries the most informative samples for model selection. Experimental results on both toy and real-world datasets show the effectiveness of *AMS* even with only few samples labeled.

6. ACKNOWLEDGEMENT

This work is supported by the project (60675009) of the National Natural Science Foundation of China.

7. REFERENCES

[1] X. Zhu, "Semi-supervied learning literature survey," *Computer Sciences Technical Report, 1530, University of Wisconsin-Madison*, 2006.

[2] O. Chapelle, B. Scholkopf, and A. Zien, *Semisupervised Learning*, MIT Press: Cambridge, MA, 2006.

[3] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems*, 2004, vol. 16.

[4] O. Chapelle, J. Weston, and B. Scholkopf, "Cluster kernels for semi-supervised learning," in *Advances in Neural Information Processing Systems*, 2003, vol. 15.

[5] X. Zhu, *Semi-Supervised Learning with Graphs*, Doctoral thesis, Carnegie Mellon University, May 2005.

[6] O. Delalleu, Y. Bengio, and N. Le Roux, "Non-parametric function induction in semi-supervised learning," in *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.

[7] F. Chung, *Spectral Graph Theory*, American Mathematical Society, 1997.

[8] S. Kirkpatrick, C. D. Gelatt, and Jr. M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[9] Xinhua Zhang and Wee Sun Lee, "Hyperparameter learning for graph based semi-supervised learning algorithms," in *Advances in Neural Information Processing Systems*, 2007, vol. 19.

[10] Ashish Kapoor, Yuan (Alan) Qi, Hyungil Ahn, and Rosalind W. Picard, "Hyperparameter and kernel learning for graph based semi-supervised classification," in *Advances in Neural Information Processing Systems*, 2006, vol. 18.