Towards Active Learning on Graphs: An Error Bound Minimization Approach

Quanquan Gu and Jiawei Han
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL, 61820
qgu3@illinois.edu, hanj@cs.uiuc.edu

***Abstract*—Active learning on graphs has received increasing interest in the past years. In this paper, we propose a *nonadaptive* active learning approach on graphs, based on generalization error bound minimization. In particular, we present a data-dependent error bound for a graph-based learning method, namely learning with local and global consistency (LLGC). We show that the empirical transductive Rademacher complexity of the function class for LLGC provides a natural criterion for active learning. The resulting active learning approach is to select a subset of nodes on a graph such that the empirical transductive Rademacher complexity of LLGC is minimized. We propose a simple yet effective sequential optimization algorithm to solve it. Experiments on benchmark datasets show that the proposed method outperforms the state-of-the-art active learning methods on graphs.**

***Keywords*-Active Learning; Graph; Generalization Error Bound; Sequential Optimization**

I. Introduction

In many practical machine learning problems, the acquisition of labeled data is often expensive and/or time consuming. This motivates *Active Learning* [7], which attempts to select the most informative data points for labeling to reduce the labeling cost. Traditional active learning methods [15] [16] [12] focus on the data which are represented by vectors. However, in many real-life applications, the data are represented by a graph, e.g., bibliographic networks. Moreover, the data which are represented by vectors can be transformed into a graph by standard techniques widely used in graph-based semi-supervised learning [18] [17]. Therefore, active learning on graphs is an alternative of practical interest to traditional active learning and has received increasing attention.

Depending on whether there is an interaction between the learner and the oracle, active learning can be roughly categorized into two families. One is *adaptive* active learning, such as SVM active learning [15], and agnostic active learning [1], which is able to use previous labels to determine the next point to label. The other family is *nonadaptive* active learning, which is appealing because it is able to select a batch of data points without training a classifier. For example, optimal experimental design methods [16] [12] have been used for nonadaptive active learning. In our study, we mainly focus on *nonadaptive* active learning.

In recent years, many active learning methods on graphs have been proposed, motivated by different criteria of "informative" data. For example, [11] derived a deterministic error bound for *Minimum Cut*-based semi-supervised learning approach [4], which shows that the prediction error is small if the graph cut size is large. This suggests a label selection method to choose the labeled nodes to maximize the graph cut size. Therefore, they proposed a heuristic algorithm to maximize the graph cut for active learning. [10] generalized the error bound in [11] by replacing the graph cut with an arbitrary symmetric submodular function, and also proposed an improved algorithm to maximize the graph cut using submodular function maximization technique. [11] proposed a probabilistic error bound that motivates an active learning method, which first clusters the graph and then randomly chooses a node in each cluster. [13] proposed to select the most informative nodes by minimizing the prediction variance of *Gaussian Filed and Harmonic Function* (GFHF) [18]. All the active learning methods mentioned above are non-adaptive. Another line of research [5] [3] has considered adaptive active learning, where the labels for the nodes of a graph are queried and predicted in an iterative way.

In this paper, we aim to develop a non-adaptive active learning method on graphs, which theoretically guarantees a good generalization performance. To achieve this goal, it is natural to consider the generalization error of a specific classifier on graphs. In particular, we choose *Learning with Local and Global Consistency* (LLGC) [17] as the classifier on graphs, because it is comparable to or even better than *Minimum Cut* (MinCut) [4] and GFHF [18]. We present a data-dependent generalization error bound for LLGC using the tool of transductive Rademacher Complexity [8], which is an extension of inductive Rademacher Complexity [2] and measures the richness of a class of real-valued functions with respect to a probability distribution. We show that the empirical transductive Rademacher complexity is a good surrogate for active learning on graphs. Thus we propose to actively select the nodes by minimizing the empirical transductive Rademacher complexity of LLGC on a graph. The resulting active learning method is a combinatorial optimization problem. In order to optimize it effectively, we present a sequential optimization algorithm. It is worth noting that our proposed active learning method tends to result in small generalization error for LLGC[1]. Experiments on benchmark datasets show that the proposed method outperforms the state-of-the-art active learning methods on graphs.

The remainder of this paper is organized as follows. In Section II, we present a generalization error bound for LLGC. In Section III, we present a criterion for active learning and its optimization algorithm. The experiments are demonstrated in Section IV. Finally, we draw conclusions and point out some future work in Section V.

II. Analysis of Learning with Local and Global Consistency

Given a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v_i \in \mathcal{V}$ corresponds to a data point $\mathbf{x}_i$, and the weight W_{ij} of edge $e_{ij} \in \mathcal{E}$ reflects the affinity between i-th node and the j-th node. $\mathbf{W} \in \mathbb{R}^{n\times n}$ is called adjacency matrix of the graph. For undirected graphs, $\mathbf{W}$ is symmetric, while for directed graphs, $\mathbf{W}$ is asymmetric. In the setting of classification, some of the nodes on the graph are labeled, i.e., $y_i \in \{\pm 1\}$, while the remainder are unlabeled, i.e., $y_i = 0$. And our goal is to predict the labels of those unlabeled nodes.

A. Review of LLGC

There exist bunches of graph-based (semi-supervised) learning methods, e.g., Minimum Cut (MinCut) [4], Gaussian Field and Harmonic Function (GFHF) [18] and Learning with Local and Global Consistency (LLGC) [17]. In this paper, we focus on LLGC because it is the state-of-the-art method and amenable to theoretical analysis.

In order to preserve the topological properties of a graph, LLGC [17] assumes that if two nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected in the graph, then the labels of these two nodes tend to be similar to each other. Given a symmetric adjacency matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ of the graph, and let $f(\mathbf{x}_i)$ be the label of node $\mathbf{x}_i$ produced by a classifier f, the above assumption can be mathematically formulated as:

$$\frac{1}{2}\sum_{i,j=1}^{n}(\frac{f_i}{\sqrt{D_{ii}}} - \frac{f_j}{\sqrt{D_{jj}}})^2 W_{ij} = \mathbf{f}^T\mathbf{L}\mathbf{f}, \tag{1}$$

where f_i is a shorthand for $f(\mathbf{x}_i)$, $\mathbf{f} = [f_1, \ldots, f_n]^T$, $\mathbf{D}$ is a diagonal matrix, called degree matrix, with $D_{ii} = \sum_{j=1}^{n} W_{ij}$, $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$ is the normalized graph Laplacian [6], and $\mathbf{I}$ is an identity matrix of appropriate size. Eq. (1) is called *Graph Regularization*. Intuitively, the objective function incurs a heavy penalty if neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped far apart.

In the setting of binary classification, LLGC pursues a function f by minimizing the following criterion

$$\min_{\mathbf{f}} ||\mathbf{f} - \mathbf{y}||_2^2 + \mu \mathbf{f}^T\mathbf{L}\mathbf{f}, \tag{2}$$

where $\mu > 0$ is a regularization parameter, which controls the balance between the loss and label smoothness. $\mathbf{y}$ is the label vector with $\mathbf{y} = [y_1, y_2, \ldots, y_n]^T$.

B. Generalization Error Bound for LLGC

In this subsection, we derive a generalization error bound for LLGC using the tool of transductive Rademacher complexity for general function classes [8].

Definition 1. *[8] For a fixed sample set $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ generated by a distribution $\mathcal{D}_{\mathcal{X}}$ on a set $\mathcal{X}$ and a real-valued function class $\mathcal{F}$ with domain $\mathcal{X}$, the empirical transductive Rademacher complexity of $\mathcal{F}$ is the random variable*

$$\hat{R}_{l+u}(\mathcal{F}) = (\frac{1}{l} + \frac{1}{u})\mathbb{E}_{\sigma}\left[\sup_{f\in\mathcal{F}}\sum_{i=1}^{l+u}\sigma_i f(\mathbf{x}_i)\right], \tag{3}$$

where $l + u = n$, and $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_n)^T$ are independent random variables such that

$$\sigma_i = \begin{cases} 1 & w.p.\ p \\ -1 & w.p.\ p \\ 0 & w.p.\ 1-2p, \end{cases} \tag{4}$$

where $0 \le p \le \frac{1}{2}$. The transductive Rademacher complexity is

$$R_{l+u}(\mathcal{F}) = \mathbb{E}_{\mathbf{x}}\left[\hat{R}_{l+u}(\mathcal{F})\right]. \tag{5}$$

Note that for the case $p = \frac{1}{2}$ and $l = u$, the transductive Rademacher complexity coincides with the standard inductive definition [2] up to a normalization factor $\frac{1}{l} + \frac{1}{u}$. We set $p = \frac{lu}{n^2}$ in the following derivation.

Intuitively speaking, transductive Rademacher complexity measures the richness of a class of real-valued functions with respect to a probability distribution.

Theorem 2. *[8] Fix $\delta \in (0, 1)$, and let $\mathcal{F}$ be a class of functions mapping from $\mathcal{X} \times \mathcal{Y}$ to $[0, 1]$. Let $c_0 = \sqrt{\frac{32\ln(4e)}{3}}$ and $Q = \frac{1}{l} + \frac{1}{u}$. For any fixed sample set $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, with probability $1 - \delta$ over random draws of a subsample of size l, every $f \in \mathcal{F}$ satisfies*

$$\begin{aligned} err(f) &\le \hat{err}(f) + \hat{R}_{l+u}(\mathcal{F}) \\ &+ c_0 Q\sqrt{min(l,u)} + \sqrt{2Q\ln(1/\delta)}, \end{aligned} \tag{6}$$

where $err(f)$ is the expected error on the unlabeled data, and $\hat{err}(f)$ is the empirical error on the labeled data.

The above error bound is quite general and applicable to various transductive learning algorithms if an empirical transductive Rademacher complexity $\hat{R}_{l+u}(\mathcal{F})$ of the function class $\mathcal{F}$ can be found efficiently. It also implies that in order to prove the generalization error bound for LLGC, it is sufficient to give an estimation of the empirical transductive Rademacher complexity for the following function class.

Definition 3. *The function class of LLGC is $\mathcal{F}_l = \{\mathbf{f} = (\mu\mathbf{L} + \mathbf{I})^{-1}\mathbf{y}, ||\mathbf{y}||_2 \le \sqrt{l}\}$.*

[1] Although it is designed based on LLGC, we will show that it also works very well for GFHF by experiments.

Note that there are l labeled data with $y_i \in \{\pm 1\}$, therefore, $\|\mathbf{y}\|_2 \leq \sqrt{l}$.

In the following, we present two theorems, which serve as the theoretical foundation of our proposed method in this paper.

Theorem 4. *The empirical transductive Rademacher complexity of the function class $\mathcal{F}_l$ is upper bounded as*

$$\hat{R}_{l+u}(\mathcal{F}_l) \quad \leq \quad \sqrt{\frac{2}{u} tr\left((\mu\mathbf{L}+\mathbf{I})^{-2}\right)}, \tag{7}$$

where $\mathbf{I}$ *is an identity matrix.*

Proof: The empirical Rademacher complexity of the function class $\mathcal{F}_l$ is computed as

$$\begin{aligned} & \hat{R}_{l+u}(\mathcal{F}_l) \\ = \ & (\frac{1}{l}+\frac{1}{u})\mathbb{E}_\sigma\left[\sup_{\mathbf{f}:||\mathbf{y}||_2\leq\sqrt{l}} \mathbf{y}^T(\mu\mathbf{L}+\mathbf{I})^{-1}\boldsymbol{\sigma}\right] \\ \leq \ & (\frac{1}{l}+\frac{1}{u})\mathbb{E}_\sigma\left[\sup_{\mathbf{f}:||\mathbf{y}||_2\leq\sqrt{l}} ||\mathbf{y}||_2||(\mu\mathbf{L}+\mathbf{I})^{-1}\boldsymbol{\sigma}||_2\right] \\ \leq \ & (\frac{1}{l}+\frac{1}{u})\sqrt{l}\mathbb{E}_\sigma\left[\sqrt{\sum_{i,j=1}^{n}\sigma_i\sigma_j\left((\mu\mathbf{L}+\mathbf{I})^{-2}\right)_{ij}}\right] \\ \leq \ & (\frac{1}{l}+\frac{1}{u})\sqrt{l}\sqrt{\frac{2lu}{n^2}\mathrm{tr}\left((\mu\mathbf{L}+\mathbf{I})^{-2}\right)} \\ = \ & \sqrt{\frac{2}{u}\mathrm{tr}\left((\mu\mathbf{L}+\mathbf{I})^{-2}\right)}, \end{aligned} \tag{8}$$

where the first inequality holds due to the Cauchy-Schwarz's inequality and the third inequality holds due to the Jensen's inequality. ■

Using Theorem 2 and Theorem 4, we obtain the following generalization error bound for LLGC.

Theorem 5. *Fix $\delta \in (0,1)$, and let $\mathcal{F}$ be a class of functions mapping from $\mathcal{X}\times\mathcal{Y}$ to $[0,1]$. Let $c_0 = \sqrt{\frac{32\ln(4e)}{3}}$ and $Q = \frac{1}{l}+\frac{1}{u}$. For any fixed sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, with probability $1-\delta$ over random draws of a subsample of size l, every $f \in \mathcal{F}$ satisfies*

$$\begin{aligned} err(f) \quad & \leq \quad \hat{err}(f) + \sqrt{\frac{2}{u} tr\left((\mu\mathbf{L}+\mathbf{I})^{-2}\right)} \\ & + \quad c_0 Q\sqrt{min(l,u)} + \sqrt{2Q\ln(1/\delta)}. \end{aligned} \tag{9}$$

III. Active Learning via Error Bound Minimization

The generic problem of non-adaptive active learning on graphs is as follows. Given a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V}$ is the pool of candidate nodes, our goal is to find a subset $\mathcal{L} \subset \mathcal{V}$, which contains the most informative l nodes, namely active set or labeled set. Let $\mathcal{U} = \mathcal{V} \setminus \mathcal{L}$ be the unlabeled set. Given a graph Laplacian matrix $\mathbf{L}$ associated with the graph, $\mathbf{L}_{\mathcal{L}\mathcal{L}}$ denotes the principal submatrix corresponding to the labeled set $\mathcal{L}$, $\mathbf{L}_{\mathcal{U}\mathcal{U}}$ denotes the principal submatrix corresponding to the unlabeled set $\mathcal{U}$, and $\mathbf{L}_{\mathcal{L}\mathcal{U}}$ denotes the submatrix which interrelates the labeled set $\mathcal{L}$ with unlabeled set $\mathcal{U}$.

A. Objective Function

From Theorem 5, we can see that the expected error on the unlabeled data is upper bounded by the empirical error on the labeled data plus the empirical transductive Rademacher complexity $\hat{R}_{l+u}(\mathcal{F}_l)$ and the confidence term $c_0 Q\sqrt{min(l,u)} + \sqrt{2Q\ln(1/\delta)}$. It is easy to check that, the larger the number of labeled samples (l) is, the tighter the bound will be. In other words, the expected error on the unlabeled data will be approximated by the empirical error more accurately. Ideally we should minimize the expected error on the unlabeled data by jointly minimizing the empirical error on the labeled data and $\hat{R}_{l+u}(\mathcal{F}_l)$. However, in the setting of non-adaptive active learning on a graph, we do not know the label of a given node until we select this node. That means we cannot estimate $\hat{err}(f)$ before we label the nodes and train a classifier. Hence the only term we can control is the empirical transductive Rademacher complexity. In other words, minimizing $\hat{R}_{l+u}(\mathcal{F}_l)$ is a surrogate to guarantee small expected error. Therefore, we present an active learning criterion by minimizing the upper bound of empirical transductive Rademacher complexity for LLGC as follows,

$$\arg\min_{\mathcal{L}\subset\mathcal{V}} \mathrm{tr}\left((\mu\mathbf{L}_{\mathcal{L}\mathcal{L}}+\mathbf{I})^{-2}\right). \tag{10}$$

where we ignore the constant scalers and square root symbol. Note that $\mathbf{L}_{\mathcal{L}\mathcal{L}}$ is computed based on the selected l samples, i.e., $\mathcal{L}$.

The above optimization problem is a combinatorial optimization problem. Finding the global optimal solution is NP-hard. Motivated by the success of sequential minimization algorithm in some existing experimental design approaches [16] [12] [13], we present a simple yet effective sequential optimization algorithm as follows.

B. Sequential Optimization

We introduce a selection matrix $\mathbf{S} \in \mathbb{R}^{n\times l}$, which is defined as

$$S_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \text{ is selected as the } j\text{-th point in } \mathcal{L} \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to check that each column of $\mathbf{S}$ has one and only one 1, each row has at most one 1, and $\mathbf{S}^T\mathbf{S} = \mathbf{I}$. The constraint set for $\mathbf{S}$ can be defined as

$$\begin{aligned} \mathcal{S} \quad & = \quad \{\mathbf{S}|\mathbf{S}\in\{0,1\}^{n\times l}, \mathbf{S}^T\mathbf{1} = \mathbf{1}, \mathbf{S}\mathbf{1} \leq \mathbf{1}\} \\ & = \quad \{\mathbf{S}|\mathbf{S}\in\{0,1\}^{n\times l}, \mathbf{S}^T\mathbf{S} = \mathbf{I}\}, \end{aligned} \tag{11}$$

Then $\mathbf{L}_{\mathcal{L}\mathcal{L}}$ can be represented by $\mathbf{S}^T\mathbf{L}\mathbf{S}$. Therefore, Eq. (10) can be equivalently written as

$$\arg\min_{\mathcal{S}\subset\mathcal{S}} \mathrm{tr}\left((\mu\mathbf{S}^T\mathbf{L}\mathbf{S}+\mathbf{I})^{-2}\right). \tag{12}$$

Suppose the eigen decomposition of $\mathbf{L}$ is $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ where $\mathbf{U}$ is consisted of the eigenvectors, and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_n)$ is a diagonal matrix whose diagonal elements are eigenvalues. We have

$$\begin{aligned}\text{tr}\left((\mu\mathbf{S}^T\mathbf{L}\mathbf{S}+\mathbf{I})^{-2}\right) &= \text{tr}\left((\mu\mathbf{S}^T\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T\mathbf{S}+\mathbf{I})^{-2}\right)\\ &= \text{tr}\left((\mathbf{S}^T\mathbf{U}(\mu\mathbf{\Lambda}+\mathbf{I})\mathbf{U}^T\mathbf{S})^{-2}\right)\\ &= \text{tr}\left((\mathbf{S}^T\mathbf{U}\mathbf{\Sigma}\mathbf{U}^T\mathbf{S})^{-1}\right)\\ &= \text{tr}\left((\mathbf{S}^T\mathbf{U}\mathbf{\Gamma}\mathbf{U}^T\mathbf{S}+\mathbf{I})^{-1}\right) \quad (13)\end{aligned}$$

where $\mathbf{\Sigma} = \text{diag}\left((\mu\lambda_1+1)^2, \ldots, (\mu\lambda_n+1)^2\right)$, $\mathbf{\Gamma} = \text{diag}\left((\mu\lambda_1+1)^2-1, \ldots, (\mu\lambda_n+1)^2-1\right)$, and we use the fact that $\mathbf{S}^T\mathbf{U}\mathbf{U}^T\mathbf{S} = \mathbf{I}$.

Using the Woodbury matrix identity [9], we have

$$\begin{aligned}&(\mathbf{S}^T\mathbf{U}\mathbf{\Gamma}\mathbf{U}^T\mathbf{S}+\mathbf{I})^{-1}\\ =\ &\mathbf{I}-\mathbf{S}^T\mathbf{U}(\mathbf{\Gamma}^{-1}+\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{S}. \quad (14)\end{aligned}$$

Hence

$$\begin{aligned}&\text{tr}\left((\mathbf{S}^T\mathbf{U}\mathbf{\Gamma}\mathbf{U}^T\mathbf{S}+\mathbf{I})^{-1}\right)\\ =\ &l-\text{tr}\left(\mathbf{S}^T\mathbf{U}(\mathbf{\Gamma}^{-1}+\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{S}\right)\\ =\ &l-\text{tr}((\mathbf{\Gamma}^{-1}+\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U})^{-1}\\ &(\mathbf{\Gamma}^{-1}+\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U}-\mathbf{\Gamma}^{-1}))\\ =\ &l-n+\text{tr}\left((\mathbf{\Gamma}^{-1}+\mathbf{U}^T\mathbf{S}\mathbf{S}^T\mathbf{U})^{-1}\mathbf{\Gamma}^{-1}\right), \quad (15)\end{aligned}$$

where $\mathbf{\Gamma}^{-1}$ is a diagonal matrix whose i-th diagonal element is $\frac{1}{(\mu\lambda_i+1)^2-1}$[2]. Therefore, the optimization problem in Eq. (10) is equivalent to

$$\arg\min_{\mathcal{L}\subset\mathcal{V}} \text{tr}\left((\mathbf{\Gamma}^{-1}+\mathbf{U}_{\mathcal{L}}^T\mathbf{U}_{\mathcal{L}})^{-1}\mathbf{\Gamma}^{-1}\right), \quad (16)$$

where $\mathbf{U}_{\mathcal{L}} = \mathbf{S}^T\mathbf{U}$ is a submatrix of $\mathbf{U}$. More specially, $\mathbf{U}_{\mathcal{L}}$ consists of the rows in $\mathbf{U}$ which corresponds to the selected nodes.

Let $\mathbf{H}_0 = \mathbf{\Gamma}^{-1}$. Suppose k nodes have been selected, denoted by $\mathcal{L}_k$, which correspond to $\mathbf{U}_{\mathcal{L}_k} \in \mathbb{R}^{k\times n}$. Let $\mathbf{H}_k = \mathbf{\Gamma}^{-1}+\mathbf{U}_{\mathcal{L}_k}^T\mathbf{U}_{\mathcal{L}_k}$, then the $(k+1)$-th node can be selected by solving the following optimization problem

$$i_{k+1} = \arg\min_{i\subset\mathcal{V}/\mathcal{L}_k} \text{tr}\left((\mathbf{H}_k+\mathbf{u}_i\mathbf{u}_i^T)^{-1}\mathbf{\Gamma}^{-1}\right), \quad (17)$$

where $\mathbf{u}_i$ is the transpose of the i-th row of $\mathbf{U}$ (thus a column vector).

By using the Sherman-Morrison formula [9], we have

$$(\mathbf{H}_k+\mathbf{u}_i\mathbf{u}_i^T)^{-1} = \mathbf{H}_k^{-1} - \frac{\mathbf{H}_k^{-1}\mathbf{u}_i\mathbf{u}_i^T\mathbf{H}_k^{-1}}{1+\mathbf{u}_i^T\mathbf{H}_k^{-1}\mathbf{u}_i}. \quad (18)$$

Therefore,

$$\begin{aligned}&\text{tr}\left((\mathbf{H}_k+\mathbf{u}_i\mathbf{u}_i^T)^{-1}\mathbf{\Gamma}^{-1}\right)\\ =\ &\text{tr}(\mathbf{H}_k^{-1}\mathbf{\Gamma}^{-1}) - \frac{\mathbf{u}_i^T\mathbf{H}_k^{-1}\mathbf{\Gamma}^{-1}\mathbf{H}_k^{-1}\mathbf{u}_i}{1+\mathbf{u}_i^T\mathbf{H}_k^{-1}\mathbf{u}_i}. \quad (19)\end{aligned}$$

Since $\text{tr}(\mathbf{H}_k^{-1}\mathbf{\Gamma}^{-1})$ is a constant given $\mathbf{H}_k^{-1}$, the optimization problem in Eq. (17) is equivalent to

$$i_{k+1} = \arg\max_{i\subset\mathcal{V}/\mathcal{L}_k} \frac{\mathbf{u}_i^T\mathbf{H}_k^{-1}\mathbf{\Gamma}^{-1}\mathbf{H}_k^{-1}\mathbf{u}_i}{1+\mathbf{u}_i^T\mathbf{H}_k^{-1}\mathbf{u}_i}. \quad (20)$$

Once the $(k+1)$-th node is selected, $\mathbf{H}_{k+1}^{-1}$ can be updated based on $\mathbf{H}_k^{-1}$, by using the Sherman-Morrison formula again,

$$\mathbf{H}_{k+1}^{-1} = \mathbf{H}_k^{-1} - \frac{\mathbf{H}_k^{-1}\mathbf{u}_{i_{k+1}}\mathbf{u}_{i_{k+1}}^T\mathbf{H}_k^{-1}}{1+\mathbf{u}_{i_{k+1}}^T\mathbf{H}_k^{-1}\mathbf{u}_{i_{k+1}}}. \quad (21)$$

Note that $\mathbf{H}_{k+1}^{-1}$ is updated by matrix (vector) multiplication and addition, rather than matrix inverse. Therefore, this process is efficient.

In summary, we present the whole algorithm for active learning on graphs in Algorithm 1.

Algorithm 1 Active Learning on Graphs via Generalization Error Bound Minimization (**Bound**)

```
Input: Adjacency matrix $\mathbf{W}$, number of nodes to select $l$, regularization parameter $\mu$;
Compute $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$
Perform eigen decomposition $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$
Initialize $\mathbf{H}_0 = \mathbf{\Gamma}^{-1}$, $\mathcal{L}_0 = \emptyset$
for $k = 0 \to l-1$ do
    Compute $i_{k+1} = \arg\max_{i\subset\mathcal{V}/\mathcal{L}_k} \frac{\mathbf{u}_i^T\mathbf{H}_k^{-1}\mathbf{\Lambda}^{-1}\mathbf{H}_k^{-1}\mathbf{u}_i}{1+\mathbf{u}_i^T\mathbf{H}_k^{-1}\mathbf{u}_i}$;
    Update $\mathcal{L}_{k+1} = \mathcal{L}_k \cup \{i_{k+1}\}$
    Update $\mathbf{H}_{k+1}^{-1} = \mathbf{H}_k^{-1} - \frac{\mathbf{H}_k^{-1}\mathbf{u}_{i_{k+1}}\mathbf{u}_{i_{k+1}}^T\mathbf{H}_k^{-1}}{1+\mathbf{u}_{i_{k+1}}^T\mathbf{H}_k^{-1}\mathbf{u}_{i_{k+1}}}$
end for
```

C. Complexity Analysis

The computational complexity of Algorithm 1 includes two parts. The first part is eigen-decomposition of the adjacency matrix $\mathbf{W}$. For a graph whose average node degree is k, the Lanczos algorithm [9] can be used to efficiently compute the eigenvectors of the eigen-problem within $O(tn^2k)$, where t is the number of iterations in Lanczos. The second part is the sequential optimization algorithm, whose complexity is $O(n^2l)$ where l is the number of selected nodes, i.e., $|\mathcal{L}|$. Hence the total time complexity is $O(n^2(tk+l))$, which is applicable to medium-scale graphs.

IV. Experiments

In this section, we evaluate the proposed method on real-world datasets, and compare it with the state-of-the-art active learning methods on graphs. Recall that the input of active learning methods on graphs is an adjacency matrix.

[2] Since the smallest eigenvalue of graph Laplacian is 0, $\mathbf{\Gamma}^{-1}$ is ill-defined. In our implementation, we resolve this problem by replacing the zero eigenvalue with a sufficient small value, e.g., $1e-6$.

A. Datasets

In our experiments, we use three real-world benchmark datasets to evaluate the active learning methods.

Cora contains the abstracts and references of about $34,000$ research papers from the computer science community. The task is to classify each paper into one of the subfields of data structure (DS), hardware and architecture (HA), machine learning (ML), and programming language (PL), based on the citation relation between the papers. We only use the link information of this dataset. We choose DS and PL subsets to form two datasets. For each dataset, the largest connected component of the graph is used. Since the adjacency matrix of the Cora dataset is directed, we symmetrize it by $\max(\mathbf{W}, \mathbf{W}^T)$.

Coauthor is an undirected co-author graph data extracted from the DBLP[3] database in four areas: machine learning, data mining, information retrieval and database. It contains a total of 1711 authors, each of which is represented by a node. The edge between each pair of authors is weighted by the number of papers they co-authored. Each class contains about 400 authors. This graph is already connected.

B. Methods & Parameter Settings

To demonstrate the effectiveness of our proposed method, we compare the following active learning approaches.

- Random Sampling (**Random**) uniformly selects nodes from the candidate set. It is the simplest baseline for active learning.
- Variance Minimization (**VM**) [13] is a recently proposed method, which is motivated by GFHF and minimizing the prediction variance.
- **METIS** [11]: it uses the METIS clustering method [14] to divide a graph into l clusters, and randomly chooses one data point from each cluster.
- Ψ Maximization (**Ψ-Max**) [10]: it is solved by submodular function maximization, which performs better than the heuristic optimization algorithm proposed in [11].
- Generalization Error Bound Minimization (**Bound**) is our proposed method. It is motivated by Theorem 5. There is one parameter μ tunable. Throughout our experiments, we simply fix $\mu = 0.01$.

After selecting the nodes by active learning, we train a classifier on the graph to do classification. In our experiments, we tried three classifiers: LLGC, GFHF and MinCut. The reason why we tried these three classifiers is obvious, because the proposed active learning method is built upon LLGC, VM is motivated by GFHF and Ψ-Max is designed for MinCut. There is a parameter for LLGC, i.e., μ, which is tuned by 3-fold cross validation on the selected labeled set over the grid $\{0.01, 0.1, 1, 10, 100\}$.

C. Experimental Setup

In order to randomize the experiments, in each run of experiments, we restrict the pool of the candidate nodes to be selected from a random sampling of 50% of the total nodes. The random split was repeated 10 times. For each dataset, we let the active learning methods incrementally choose $\{10, 20, \dots, 160\}$ nodes from the training set to label. We evaluate different active learning methods combined with different classifiers. We compute the mean classification accuracy on all the unlabeled nodes, that is, the unselected nodes in the pool plus the remaining 50% nodes.

D. Classification Results

The experimental results evaluated on the unlabeled data are shown in Figures 1 and 2. In all subfigures, the horizontal axis represents the number of labeled nodes, while the vertical axis is the averaged classification accuracy over 10 runs. The experimental result of MinCut is much worse than LLGC and GFHF for all the active learning methods, because it usually results in a very unbalanced classification. Therefore, we omit its result.

From Figures 1 and 2, we observe that the proposed method (Bound) consistently outperforms other methods in most cases using either LLGC or GFHF. It is appealing because even though our method is built upon the error bound minimization of LLGC, it is also much better than other methods using GFHF. But note that our method using LLGC achieves marginally better performance than using GFHF. On the Cora datasets, when the number of labeled nodes is small, e.g., less than 30, our method and Ψ-Max usually perform the best. On the other cases, our method is much better than the second best method. The superior performance of our method is attributed to its theoretical foundation, which guarantees that the classifier can achieve small generalization error on the unlabeled data.

VM is usually worse than random sampling. The reason is that minimizing the prediction variance does not guarantee the quality of predictions on the unlabeled data.

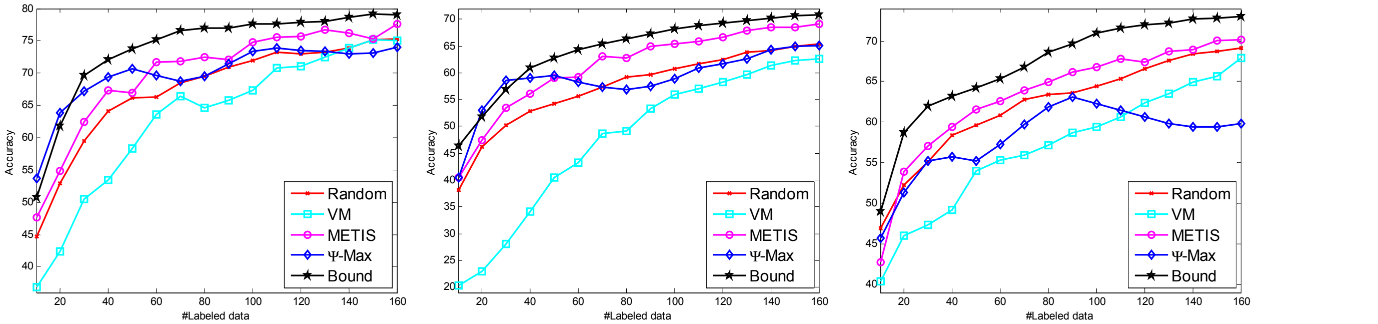
The performance of METIS is usually comparable to or even better than that of Ψ-Max. Although METIS has a solid theoretical foundation, the corresponding criterion is so difficult that we have to solve it by a clustering algorithm [14] followed by a heuristic sampling, which sacrifices its performance.

In summary, our method together with LLGC is the most promising combination, which is consistent with our theory.

V. Conclusions and Future Work

The main contributions of this paper are: (1) We present a generalization error bound for LLGC; (2) we present an active learning criterion for graph data via minimizing the empirical transductive Rademacher complexity of LLGC; and (3) we present a simple algorithm to optimize the active

[3] www.informatik.uni-trier.de/ ley/db/

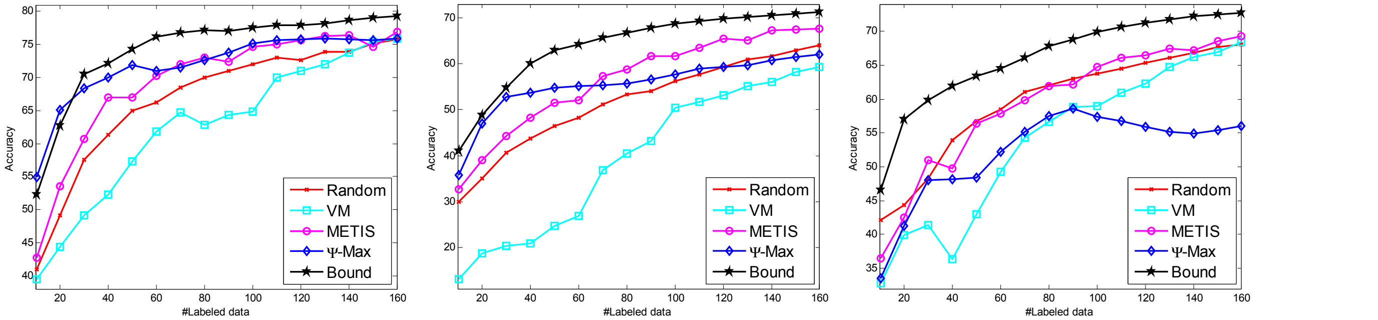


(a) Cora (DS)

(b) Cora (PL)

(c) Coauthor

Figure 1. Comparison of active learning methods on (a) Cora (DS); (b) Cora (PL); and (c) Coauthor using LLGC evaluated on all the unlabeled data.



(a) Cora (DS)

(b) Cora (PL)

(c) Coauthor

Figure 2. Comparison of active learning methods on (a) Cora (DS); (b) Cora (PL); and (c) Coauthor using GFHF evaluated on all the unlabeled data.

learning criterion. In the future, we plan to develop a more scalable algorithm to solve Eq. (10).

Acknowledgement

The work was supported in part by U.S. National Science Foundation grants IIS-0905215, CNS-0931975, the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NS-CTA), the U.S. Air Force Office of Scientific Research MURI award FA9550-08-1-0265, and MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC.

References

[1] M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *ICML*, pages 65–72, 2006.

[2] P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[3] M. Bilgic, L. Mihalkova, and L. Getoor. Active learning for networked data. In *ICML*, pages 79–86, 2010.

[4] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *ICML*, pages 19–26, 2001.

[5] N. Cesa-Bianchi, C. Gentile, F. Vitale, and G. Zappella. Active learning on trees and graphs. In *COLT*, pages 320–332, 2010.

[6] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, February 1997.

[7] D. A. Cohn, L. E. Atlas, and R. E. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[8] R. El-Yaniv and D. Pechyony. Transductive rademacher complexity and its applications. *J. Artif. Intell. Res. (JAIR)*, 35:193–234, 2009.

[9] G. H. Golub and C. F. V. Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, 1996.

[10] A. Guillory and J. Bilmes. Active semi-supervised learning using submodular functions. In *UAI*, pages 274–282, 2011.

[11] A. Guillory and J. A. Bilmes. Label selection on graphs. In *NIPS*, pages 691–699, 2009.

[12] X. He, W. Min, D. Cai, and K. Zhou. Laplacian optimal design for image retrieval. In *SIGIR*, pages 119–126, 2007.

[13] M. Ji and J. Han. A variance minimization criterion to active learning on graphs. *AISTATS*, pages 556–564, 2012.

[14] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, Dec. 1998.

[15] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *ICML*, pages 999–1006, 2000.

[16] K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In *ICML*, pages 1081–1088, 2006.

[17] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, 2003.

[18] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.