A Variance Minimization Criterion to Active Learning on Graphs

Ming Ji
Department of Computer Science
University of Illinois at Urbana-Champaign

Jiawei Han
Department of Computer Science
University of Illinois at Urbana-Champaign

Abstract

We consider the problem of active learning over the vertices in a graph, without feature representation. Our study is based on the common graph smoothness assumption, which is formulated in a Gaussian random field model. We analyze the probability distribution over the unlabeled vertices conditioned on the label information, which is a multivariate normal with the mean being the harmonic solution over the field. Then we select the nodes to label such that the total variance of the distribution on the unlabeled data, as well as the expected prediction error, is minimized. In this way, the classifier we obtain is theoretically more robust. Compared with existing methods, our algorithm has the advantage of selecting data in a batch offline mode with solid theoretical support. We show improved performance over existing label selection criteria on several real world data sets.

1 Introduction

In many domains of interest, data instances are connected by edges representing certain relationships, forming a graph structure. Graphs and feature vectors are two alternatives to represent the data, and the former is often more natural than the latter in many data sets [9] including people linked by the friendship relation in social networks, web pages interconnected by hyperlinks, etc. Even if the original data has feature representation, it is usually helpful to transform the data into a graph structure (via constructing a nearest neighbor graph, for instance) to better exploit properties of the data. In this way, learning on graphs is receiving more and more attention in recent years.

Substantial efforts have been devoted to the problem of classification of the nodes in a graph. On the other hand, labels can be very expensive to obtain in many real-world applications. Active learning [7] is then proposed to determine which data examples should be labeled such that the classifier could achieve higher prediction accuracy over the unlabeled data as compared to random label selection. The goal of active learning is to maximize the learner's ability given a fixed budget of labeling effort. While many effective active learners have been developed in literature [20], active learning that takes direct advantage of the graph structure in the data has not been explored until recently [9, 2, 25]. As large-scale data sets with inherent graph structures become increasingly prevalent, reasonable and natural active learning criteria on graphs are in great demand.

Most of the existing active learners work with data represented by feature vectors [20]. In a seminal paper [12], X. He proposes the first manifold-based active learning algorithm, i.e., LapRDD, which takes into account both the discriminant and geometrical structure in the data. A nearest neighbor graph is constructed to model the intrinsic manifold structure and incorporated into a least squares loss function as a regularizer. The most informative data points are selected by minimizing the size of the parameter covariance matrix. This principle has been successfully applied to image retrieval [12], video indexing [22], and feature selection [13]. Please see [2] for another active learning approach that exploits the features together with the graph structure. However, in some cases, features of the graph nodes are not always available. Some other methods try to select data based on the graph structure and some labeled nodes. Existing approaches have considered selecting the data that the current classifier is the most uncertain [17], the data with maximum expected information gain [23] or maximum expected entropy reduction [16]. Based on the Gaussian random field model [24], an empirical risk minimization framework [25] is proposed to select ex-

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX.

amples that minimize the empirical risk estimated by the current classifier. One major limitation of these methods [25, 17, 14, 23, 16, 15] is that they have to obtain the labels of the selected nodes in order to select more data, therefore are not applicable when there is no label information provided during active learning. When labeling an instance requires time consuming and expensive experiments, these methods are much more costly than running a batch offline mode active learner once and perform labeling in parallel [9].

Recently, there are some efforts devoted to designing label selection criteria that use the graph structure only, without feature representation and label information. Intuitively, one tends to select nodes that lie in high-density (unlabeled) regions [15] or the centers of clusters [17], or have high impact (measured by the graph structure) to unlabeled data [21]. However, these intuitive selection criteria do not have theoretical support on optimizing any classifier.

In this paper, we propose a novel variance minimization perspective to active learning purely on the graph structure, without feature representation and label information. Our study is based on the common assumption that the labels vary smoothly with respect to the graph, which is widely used in the graph-based semi-supervised learning literature [5, 3, 10, 19, 1]. Following one of the most popular graph-based learning frameworks [24], we formulate the smoothness assumption by a Gaussian random field over the graph nodes. Theoretical analysis indicates that the Gaussian field over the unlabeled vertices, conditioned on the labeled data, is a multivariate normal whose mean is the prediction of the harmonic Gaussian field classifier [24]. It is interesting to note that the covariance matrix of the Gaussian field over the unlabeled data is not dependent on the class labels, but only on the graph structure. In this way, we propose to select the data points to label such that the total variance of the Gaussian field over unlabeled examples, as well as the expected prediction error of the harmonic Gaussian field classifier, is minimized. Efficient computation scheme is then proposed to solve the corresponding optimization problem without introducing any additional parameter.

In fact, designing active learners on graphs aiming at minimizing the error of a particular classifier has received substantial interest recently [9, 25]. [9] provides theoretical bounds of the prediction error which are related to label smoothness over the graph, justifying the reasonableness of clustering the nodes and then randomly choose one point from each cluster. Compared with existing methods [9, 25, 2, 15], our algorithm has the advantage of directly minimizing the expected error (instead of the upper bound of the error) in a batch offline mode, through reasonably modeling the probability distribution over the graph. Therefore, we do not require the (potentially expensive) label information of the selected data and tedious retraining of the classifier repeatedly.

The rest of this paper is organized as follows. In the next section, we introduce the variance minimization perspective for active learning on graphs. Section 3 presents a sequential optimization scheme that efficiently solves our objective function. Extensive experimental results on three real-life data sets are presented in Section 4. We provide some concluding remarks as well as suggestions for future work in Section 5.

2 A Variance Minimization Criterion to Active Learning on Graphs

2.1 The Problem

We define the active learning problem on graphs as follows. Given a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ associated with a weight matrix W, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the set of data points (without feature representation) with true labels $\mathbf{y} = (y_1, \ldots, y_n)^T$, $\mathcal{E}$ is the set of edges between any two data points in $\mathcal{V}$, and $W = (w_{ij}) \in \mathbb{R}^{n \times n}$ where w_{ij} denotes the weight on the edge between two data points v_i and v_j. Our goal is to find a subset of points $\mathcal{L} = \{v_{p_1}, \ldots, v_{p_l}\} \subset \mathcal{V}$ where $\{p_i\}_{i=1}^{l} \subset \{1, \ldots, n\}$ are the indices of the points that we should label, such that the classifier learned from the labels on $\mathcal{L}$ could achieve the smallest expected prediction error on the unlabeled data, measured by $\sum_{v_i \in \mathcal{U}} (y_i - y_i^*)^2$, where $\mathcal{U} = \mathcal{V} \setminus \mathcal{L}$ and y_i^* is the predicted label for v_i.

Without loss of generality, in this paper, we assume that $\mathcal{G}$ is undirected and connected. We allow continuous labels here, and the labels are assumed to vary smoothly over the graph, i.e., $\sum_{i,j} w_{ij}(y_i - y_j)^2$ is small, which is similar to [9].

2.2 The Objective Function

Following [24, 25], the label smoothness assumption could be formulated by a Gaussian random field over the graph:

$$P(\mathbf{y}) = \frac{1}{Z_\beta} \exp(-\beta E(\mathbf{y})) \tag{1}$$

where $E(\mathbf{y}) = \frac{1}{2} \sum_{i,j} w_{ij} (y_i - y_j)^2$ is the energy function measuring the smoothness of a label assignment $\mathbf{y} = (y_1, \ldots, y_n)^T$ over the graph, β is an "inverse temperature" parameter, and Z_β is a partition function for the normalization purpose.

Without loss of generality, we can arrange the data points chosen to be labeled to be the first l instances,

i.e., $\mathcal{L} = \{v_1, \ldots, v_l\}$, and the rest $u(= n - l)$ examples $\mathcal{U} = \{v_{l+1}, \ldots, v_{l+u}\}$ are unlabeled. Based on the Gaussian random field model, and the constraint that the predictions on the labeled set are consistent with ground truth, i.e., $\mathbf{y}^*_{\mathcal{L}} = \mathbf{y}_{\mathcal{L}} = (y_1, \ldots, y_l)^T$, a standard method is to predict the labels with the highest probability (or equivalently, minimum energy) [24, 25]. Let $L = D - W$ be the graph Laplacian [6], where D is a diagonal matrix and $D_{ii} = \sum_j w_{ij}$. L can be split into 4 blocks according to the l-th row and column:

$$L = \begin{pmatrix} L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{pmatrix} \tag{2}$$

Then the prediction on the unlabeled nodes given by the harmonic Gaussian field classifier is [24]:

$$\mathbf{y}^*_{\mathcal{U}} = -L_{uu}^{-1} L_{ul} \mathbf{y}_{\mathcal{L}} \tag{3}$$

where $\mathbf{y}^*_{\mathcal{U}} = (y^*_{l+1}, \ldots, y^*_{l+u})^T$.

It can be proven that the Gaussian field, conditioned on the labeled data, is a multivariate normal: $\mathbf{y}_{\mathcal{U}} \sim \mathcal{N}(\mathbf{y}^*_{\mathcal{U}}, L_{uu}^{-1})$ [25], where $\mathbf{y}_{\mathcal{U}} = (y_{l+1}, \ldots, y_{l+u})^T$. Then we compute the expected prediction error on the unlabeled nodes as follows:

$$\begin{aligned} & \mathrm{E}\left(\sum_{v_i \in \mathcal{U}} (y_i - y_i^*)^2\right) \\ = \; & \mathrm{E}\left((\mathbf{y}_{\mathcal{U}} - \mathbf{y}^*_{\mathcal{U}})^T (\mathbf{y}_{\mathcal{U}} - \mathbf{y}^*_{\mathcal{U}})\right) \\ = \; & \mathrm{E}\left(\mathrm{Tr}\left((\mathbf{y}_{\mathcal{U}} - \mathbf{y}^*_{\mathcal{U}})(\mathbf{y}_{\mathcal{U}} - \mathbf{y}^*_{\mathcal{U}})^T\right)\right) \\ = \; & \mathrm{Tr}\left(\mathrm{E}\left((\mathbf{y}_{\mathcal{U}} - \mathbf{y}^*_{\mathcal{U}})(\mathbf{y}_{\mathcal{U}} - \mathbf{y}^*_{\mathcal{U}})^T\right)\right) \\ = \; & \mathrm{Tr}\left(\mathrm{var}(\mathbf{y}_{\mathcal{U}})\right) = \mathrm{Tr}(L_{uu}^{-1}) \end{aligned} \tag{4}$$

In order to minimize the expected error of the prediction results, we should minimize the variance of the statistical learning model [7]. Therefore, we propose to select the nodes to label by solving the following optimization problem:

$$\arg\min_{\mathcal{L} \subset \mathcal{V}} \mathrm{Tr}(L_{uu}^{-1}) \tag{5}$$

It is easy to verify that Eq. (5) is independent of the order of the examples, but only dependent on the choice of the set of the nodes that we choose *not* to label. Therefore, our objective function is well defined.

3 Efficient Optimization

Let $\{q_1, \ldots, q_u\}$ be the indices of the nodes that we choose *not* to label. Following the above discussion, our objective is to select a $u \times u$ submatrix L_{uu} of L on the intersections of the $\{q_1, \ldots, q_u\}$-th rows and columns, such that the trace of L_{uu}^{-1} is minimized. This optimization problem in Eq. (5) is challenging since the number of candidate sets for $\mathcal{L}$ is exponential in the total number of examples n. Moreover, since the number of unlabeled examples is usually huge, L_{uu} will likely be a large matrix and directly optimizing Eq. (5) based on the set of unlabeled data is very computationally expensive. In this section, we first transform the objective function so that it can be represented by the instances that we choose to *label*, and then propose an efficient sequential optimization scheme.

3.1 Formulations

We first construct a selection matrix $S \in \mathbb{R}^{u \times n}$ to help selecting L_{uu} from L as follows:

$$S_{ij} = \begin{cases} 1 & \text{if } j = q_i \\ 0 & \text{otherwise.} \end{cases}$$

Then we have:

$$L_{uu} = SLS^T \tag{6}$$

Since L is symmetric, it has the eigendecomposition result as follows:

$$L = X \Sigma X^T \tag{7}$$

such that X is an orthonormal matrix, and $\Sigma = \mathrm{diag}\{\lambda_1, \ldots, \lambda_n\}$, where $\{\lambda_i\}_{i=1}^n$ are the eigenvalues of L, and $\lambda_1 \geq \ldots \geq \lambda_n = 0$. Then

$$L_{uu} = SLS^T = SX\Sigma X^T S^T \tag{8}$$

Suppose $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$, where $\mathbf{x}_i^T$ is the i-th row of X. Let $Q = SX$, then $L_{uu} = Q\Sigma Q^T$. Since S is the selection matrix, then $Q = (\mathbf{q}_1, \ldots, \mathbf{q}_u)^T \in \mathbb{R}^{u \times n}$ consists of the $\{q_1, \ldots, q_u\}$-th rows of X. We further define two sets of vectors $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $\mathcal{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_u\}$, then our objective function in Eq. (5) is equivalent to the following:

$$\arg\min_{\mathcal{Q} \subset \mathcal{X}} \mathrm{Tr}\left((Q\Sigma Q^T)^{-1}\right) \tag{9}$$

Let I_n denote the identity matrix of size $n \times n$. By using the Woodbury formula [8], we have the following:

$$\begin{aligned} & (Q\Sigma Q^T)^{-1} \\ = \; & \left(Q(\Sigma + I_n)Q^T - QQ^T\right)^{-1} \\ = \; & \left(Q(\Sigma + I_n)Q^T - I_u\right)^{-1} \\ = \; & (-I_u)^{-1} - Q\left((\Sigma + I_n)^{-1} + Q^T(-I_u)^{-1}Q\right)^{-1} Q^T \\ = \; & -I_u - Q\left(M^{-1} - Q^TQ\right)^{-1} Q^T \end{aligned}$$

where $M = \Sigma + I_n = \mathrm{diag}\{\lambda_1 + 1, \ldots, \lambda_n + 1\}$. According to the matrix determinant lemma [11], we

have:

$$\begin{aligned}
& \det\left(M^{-1}-Q^TQ\right) \\
= \; & (-1)^n\det\left(-M^{-1}+Q^TQ\right) \\
= \; & (-1)^n\det\left(-M^{-1}\right)\det\left(I_u+Q\left(-M^{-1}\right)^{-1}Q^T\right) \\
= \; & (-1)^{2n}\det\left(M^{-1}\right)\det\left(I_u-QMQ^T\right) \\
= \; & \det\left(I_u-Q\Sigma Q^T-QI_nQ^T\right)\prod_{i=1}^{n}\frac{1}{\lambda_i+1} \\
= \; & \det\left(I_u-L_{uu}-I_u\right)\prod_{i=1}^{n}\frac{1}{\lambda_i+1} \\
= \; & \det(-L_{uu})\prod_{i=1}^{n}\frac{1}{\lambda_i+1} \qquad (10)
\end{aligned}$$

As long as $0 < u < n$ and the graph is connected, it can be easily proven that L_{uu} is invertible, and so is $M^{-1}-Q^TQ$. Recall that $\mathrm{Tr}(AB)=\mathrm{Tr}(BA)$, we further have:

$$\begin{aligned}
& \mathrm{Tr}\left((Q\Sigma Q^T)^{-1}\right) \\
= \; & -u-\mathrm{Tr}\left(Q\left(M^{-1}-Q^TQ\right)^{-1}Q^T\right) \\
= \; & -u-\mathrm{Tr}\left(\left(M^{-1}-Q^TQ\right)^{-1}Q^TQ\right) \\
= \; & -u \\
& +\mathrm{Tr}\left(\left(M^{-1}-Q^TQ\right)^{-1}\left(-Q^TQ+M^{-1}-M^{-1}\right)\right) \\
= \; & -u+\mathrm{Tr}\left(I_n-\left(M^{-1}-Q^TQ\right)^{-1}M^{-1}\right) \\
= \; & n-u-\mathrm{Tr}\left(\left(M^{-1}-Q^TQ\right)^{-1}M^{-1}\right) \\
= \; & l-\mathrm{Tr}\left(\left(M^{-1}-\sum_{i=1}^{u}\mathbf{q}_i\mathbf{q}_i^T\right)^{-1}M^{-1}\right)
\end{aligned}$$

Let $\mathcal{P}=\{\mathbf{p}_1,\ldots,\mathbf{p}_l\}=\mathcal{X}\setminus\mathcal{Q}$ be the $\{p_1,\ldots,p_l\}$-th row vectors of X that correspond to the examples that we choose to *label*, then we have:

$$\begin{aligned}
& \mathrm{Tr}\left((Q\Sigma Q^T)^{-1}\right) \\
= \; & l-\mathrm{Tr}\left(\left(M^{-1}-\sum_{i=1}^{u}\mathbf{q}_i\mathbf{q}_i^T\right)^{-1}M^{-1}\right) \\
= \; & l-\mathrm{Tr}\left(\left(M^{-1}-\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T+\sum_{i=1}^{l}\mathbf{p}_i\mathbf{p}_i^T\right)^{-1}M^{-1}\right) \\
= \; & l-\mathrm{Tr}\left(\left(M^{-1}-X^TX+\sum_{i=1}^{l}\mathbf{p}_i\mathbf{p}_i^T\right)^{-1}M^{-1}\right) \\
= \; & l-\mathrm{Tr}\left(\left(M^{-1}-I_n+\sum_{i=1}^{l}\mathbf{p}_i\mathbf{p}_i^T\right)^{-1}M^{-1}\right)
\end{aligned}$$

Let $A_0=M^{-1}-I_n$. Since the number of data points to be labeled, l, is fixed, our objective function in Eq. (9) reduces to the following:

$$\arg\max_{\mathcal{P}\subset\mathcal{X}}\mathrm{Tr}\left(\left(A_0+\sum_{i=1}^{l}\mathbf{p}_i\mathbf{p}_i^T\right)^{-1}M^{-1}\right) \qquad (11)$$

In the following, we describe an efficient sequential optimization scheme to select which nodes we should label in a graph.

3.2 Selecting the First Point

Setting $l=1$ in Eq. (11), we obtain the objective function of selecting one (or the first) data point to label:

$$\arg\max_{\mathbf{p}\in\mathcal{X}}\mathrm{Tr}\left(\left(A_0+\mathbf{p}\mathbf{p}^T\right)^{-1}M^{-1}\right) \qquad (12)$$

Usually, matrix inversion formulae in the form of $\left(A_0+\mathbf{p}\mathbf{p}^T\right)^{-1}$ can be simplified using the Sherman-Morrison formula [8]:

$$(A+\mathbf{u}\mathbf{v}^T)^{-1}=A^{-1}-\frac{A^{-1}\mathbf{u}\mathbf{v}^TA^{-1}}{1+\mathbf{v}^TA^{-1}\mathbf{u}} \qquad (13)$$

However, note that

$$\begin{aligned}
& A_0 \\
= \; & M^{-1}-I_n \\
= \; & \mathrm{diag}\left\{\frac{1}{\lambda_1+1}-1,\ldots,\frac{1}{\lambda_n+1}-1\right\} \\
= \; & \mathrm{diag}\left\{\frac{-\lambda_1}{\lambda_1+1},\ldots,\frac{-\lambda_n}{\lambda_n+1}\right\} \qquad (14)
\end{aligned}$$

is singular since the smallest eigenvalue of L (denoted as λ_n) is equal to 0. Therefore, the Sherman-Morrison formula (13) cannot be applied here. In this subsection, we derive how to select the first point to label by performing some modification of Eq. (12).

For a connected graph, it is known that all the eigenvalues of L, except λ_n, are larger than 0. The eigenvector corresponding to λ_n is a $n\times 1$ constant vector which can be denoted as $(c,\ldots,c)^T$. So any $\mathbf{p}\in\mathcal{X}$ can be represented as $\mathbf{p}=(\mathbf{v}^T,c)^T$ where $\mathbf{v}$ is a $(n-1)\times 1$ vector after removing the last element of $\mathbf{p}$. Let $B=\mathrm{diag}\left\{\frac{-\lambda_1}{\lambda_1+1},\ldots,\frac{-\lambda_{n-1}}{\lambda_{n-1}+1}\right\}\in\mathbb{R}^{(n-1)\times(n-1)}$ be the matrix after removing the last row and column of A_0, which is invertible. Hence:

$$\begin{aligned}
& A_0+\mathbf{p}\mathbf{p}^T \\
= \; & \begin{pmatrix} B & c\mathbf{v} \\ c\mathbf{v}^T & c^2 \end{pmatrix}+\begin{pmatrix} \mathbf{v} \\ 0 \end{pmatrix}\begin{pmatrix} \mathbf{v}^T & 0 \end{pmatrix} \\
= \; & \hat{B}+\hat{\mathbf{v}}\hat{\mathbf{v}}^T
\end{aligned}$$

where $\hat{B} = \begin{pmatrix} B & c\mathbf{v} \\ c\mathbf{v}^T & c^2 \end{pmatrix}$ and $\hat{\mathbf{v}} = \begin{pmatrix} \mathbf{v}^T & 0 \end{pmatrix}^T$. By doing blockwise matrix inversion, we have:

$$\hat{B}^{-1} = \begin{pmatrix} B^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{c^2(1-\mathbf{v}^T B^{-1}\mathbf{v})} \begin{pmatrix} c^2 B^{-1}\mathbf{v}\mathbf{v}^T B^{-1} & -cB^{-1}\mathbf{v} \\ -c\mathbf{v}^T B^{-1} & 1 \end{pmatrix}$$

where $B^{-1} = \text{diag}\left\{-\frac{\lambda_1+1}{\lambda_1}, \ldots, -\frac{\lambda_{n-1}+1}{\lambda_{n-1}}\right\}$. Now we can employ Eq. (13) and have:

$$\begin{aligned} & \left(A_0 + \mathbf{p}\mathbf{p}^T\right)^{-1} \\ = & \left(\hat{B} + \hat{\mathbf{v}}\hat{\mathbf{v}}^T\right)^{-1} \\ = & \hat{B}^{-1} - \frac{\hat{B}^{-1}\hat{\mathbf{v}}\hat{\mathbf{v}}^T\hat{B}^{-1}}{1+\hat{\mathbf{v}}^T\hat{B}^{-1}\hat{\mathbf{v}}} \end{aligned} \quad (15)$$

Recall that $M^{-1} = \text{diag}\left\{\frac{1}{\lambda_1+1}, \ldots, \frac{1}{\lambda_n+1}\right\}$. Therefore, $\left(A_0 + \mathbf{p}\mathbf{p}^T\right)^{-1} M^{-1}$ can be computed efficiently without matrix inversion for any given $\mathbf{p}$. We select the first data point to label that corresponds to $\mathbf{p} \in \mathcal{X}$ such that Eq. (12) is maximized.

3.3 Selecting More Points

We define:

$$A_l = A_0 + \sum_{i=1}^{l} \mathbf{p}_i \mathbf{p}_i^T \quad (16)$$

Suppose $l(\geq 1)$ data points have been selected, which correspond to the rows of X: $\{\mathbf{p}_1, \ldots, \mathbf{p}_l\} = \mathcal{P}_l \subset \mathcal{X}$, then the $(l+1)$-th instance can be selected by solving the following:

$$\mathbf{p}_{l+1} = \arg\max_{\mathbf{p}\in\mathcal{X}\setminus\mathcal{P}_l} \text{Tr}\left(\left(A_l + \mathbf{p}\mathbf{p}^T\right)^{-1} M^{-1}\right) \quad (17)$$

By using the Sherman-Morrison formula (13), we have:

$$\left(A_l + \mathbf{p}\mathbf{p}^T\right)^{-1} = A_l^{-1} - \frac{A_l^{-1}\mathbf{p}\mathbf{p}^T A_l^{-1}}{1+\mathbf{p}^T A_l^{-1}\mathbf{p}} \quad (18)$$

And A_1^{-1} can be computed using Eq. (15). Therefore:

$$\begin{aligned} & \text{Tr}\left(\left(A_l + \mathbf{p}\mathbf{p}^T\right)^{-1} M^{-1}\right) \\ = & \text{Tr}(A_l^{-1}M^{-1}) - \frac{\text{Tr}\left(A_l^{-1}\mathbf{p}\mathbf{p}^T A_l^{-1} M^{-1}\right)}{1+\mathbf{p}^T A_l^{-1}\mathbf{p}} \\ = & \text{Tr}(A_l^{-1}M^{-1}) - \frac{\text{Tr}\left(\mathbf{p}^T A_l^{-1} M^{-1} A_l^{-1}\mathbf{p}\right)}{1+\mathbf{p}^T A_l^{-1}\mathbf{p}} \\ = & \text{Tr}(A_l^{-1}M^{-1}) - \frac{\mathbf{p}^T A_l^{-1} M^{-1} A_l^{-1}\mathbf{p}}{1+\mathbf{p}^T A_l^{-1}\mathbf{p}} \end{aligned} \quad (19)$$

Since $\text{Tr}(A_l^{-1}M^{-1})$ is a constant when selecting the $(l+1)$-th data point, we choose the $(l+1)$-th point to label that corresponds to the following $\mathbf{p}_{l+1}$:

$$\mathbf{p}_{l+1} = \arg\min_{\mathbf{p}\in\mathcal{X}\setminus\mathcal{P}_l} \frac{\mathbf{p}^T A_l^{-1} M^{-1} A_l^{-1}\mathbf{p}}{1+\mathbf{p}^T A_l^{-1}\mathbf{p}} \quad (20)$$

Once $\mathbf{p}_{l+1}$ is obtained, A_{l+1} can be updated according to Eq. (18).

4 Experimental Results

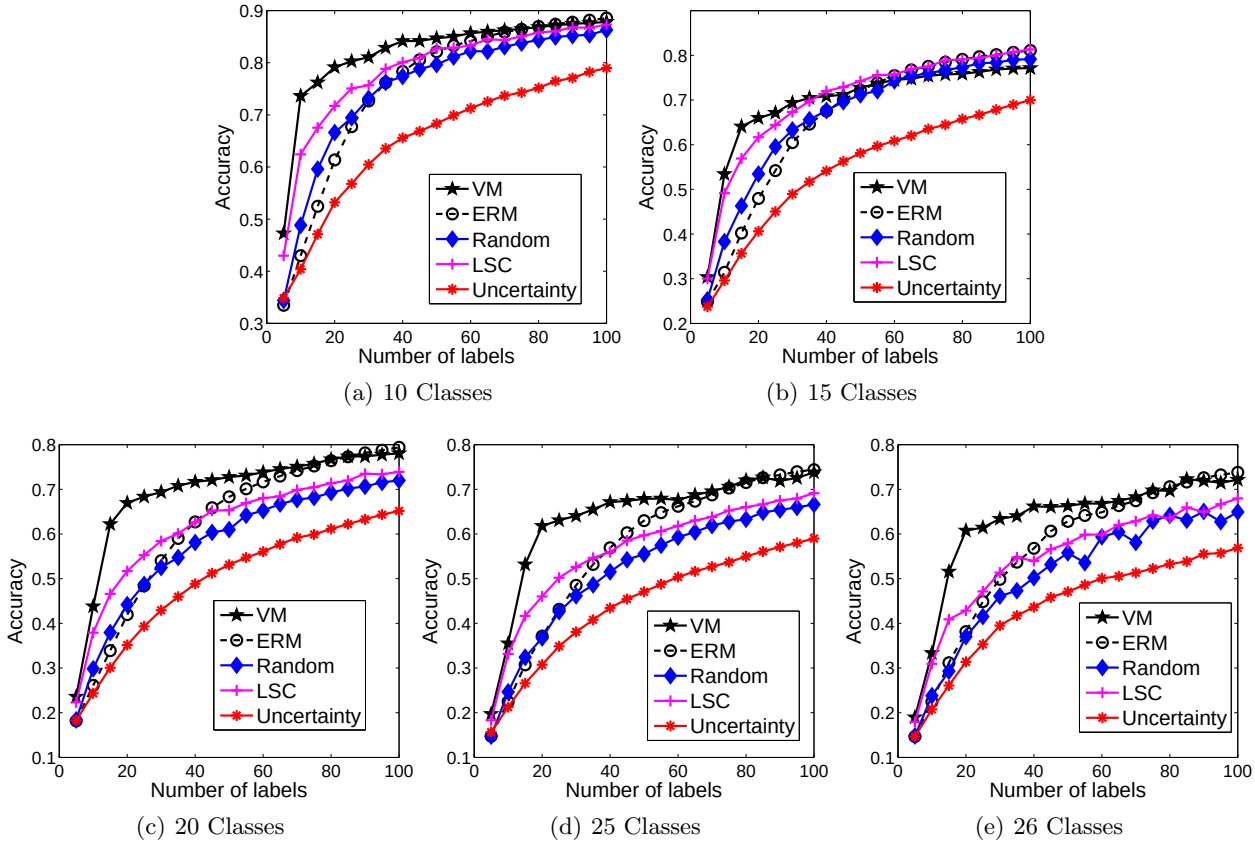
In this section, we apply our proposed active learning method based on Variance Minimization (denoted as **VM**) in the Gaussian random field to several real-world data sets to test its effectiveness. We use the labels of vertices chosen by different active learning criteria to train a harmonic Gaussian field classifier [24] to predict the labels of the rest of the nodes in the graph. The following five label selection methods are compared:

- Our proposed VM algorithm (**VM**).
- Empirical Risk Minimization (**ERM**) [4].
- Random selection (**Random**).
- Label Selection based on Clustering (**LSC**) [9].
- Uncertainty sampling (**Uncertainty**).

When our budget is to select l instances to label, the **LSC** method clusters the data into l clusters and then randomly select one example from each cluster. This method minimizes the prediction error bound related to label smoothness, and empirically performs the best in [9]. We use Spectral Clustering [18] to cluster the graph nodes. The results of **Random** and **LSC** are both averaged over 10 random trials. **ERM** and **Uncertainty** are two methods that iteratively query more data to label according to the classifier trained by the previously labeled data. **ERM** selects examples that minimize the empirical risk estimated by the current classifier. The **Uncertainty** criterion selects the instances whose labels the current classifier is the most uncertain. Recall that the harmonic Gaussian field classifier adopts the one-against-all scheme in multi-class classification. Suppose we have k classes and u unlabeled data points, then the classifier outputs a $u \times k$ score matrix, where each row is for an unlabeled point, and each column for a class. The class with the largest value in the i-th row is the predicted class of the i-th unlabeled point. Let $f_1(v_i)$ denote the largest score of node v_i related to a certain class k_1, and $f_2(v_i)$ denote the second largest score of v_i related to a different class k_2. The smaller $f_1(v_i) - f_2(v_i)$, the more

Table 1: Classification accuracy (%) by using 20 and 50 labels on the Isolet data set.

	10 classes		15 classes		20 classes		25 classes		26 classes		average	
# of labels	20	50	20	50	20	50	20	50	20	50	20	50
VM	**79.2**	**84.7**	**66.0**	72.7	**67.0**	**72.8**	**61.8**	**67.9**	**60.8**	**66.3**	**67.0**	**72.9**
ERM	61.4	82.1	48.0	72.2	42.0	68.3	37.1	63.0	38.2	62.8	45.3	69.7
Random	66.6	79.7	53.4	71.2	44.2	61.0	36.8	55.5	37.0	55.8	47.6	64.6
LSC	71.7	82.8	61.7	**74.2**	51.7	65.3	46.0	59.7	42.9	57.9	54.8	68.0
Uncertainty	53.2	68.3	40.6	58.1	35.2	53.1	30.8	47.1	31.4	47.1	38.2	54.7



(a) 10 Classes (b) 15 Classes

(c) 20 Classes (d) 25 Classes (e) 26 Classes

Figure 1: Classification accuracy vs. the number of labels used on the Isolet data set

uncertain the classifier is about the label prediction of v_i. Therefore, we select new instances $\{v_i\}$ to label with the smallest values of $f_1(v_i) - f_2(v_i)$. This strategy is also compared in [17]. Notice that **ERM** and **Uncertainty** use the label information of the previously selected data, while other active learning methods do not. In order to test them in our scenario that very little (if not none) label information is available during active learning, for **ERM** and **Uncertainty**, we randomly choose an initial set of labels for each of them, rank the other nodes according to the score of their label selection criterion (empirical risk for **ERM**, $f_1(v_i) - f_2(v_i)$ for **Uncertainty**), and select the top ranked nodes. The performance of **ERM** and **Uncertainty** are also averaged over 10 random selections of the initial set of labels.

In the following, we begin with a description of the data preparation.

4.1 Data Preparation

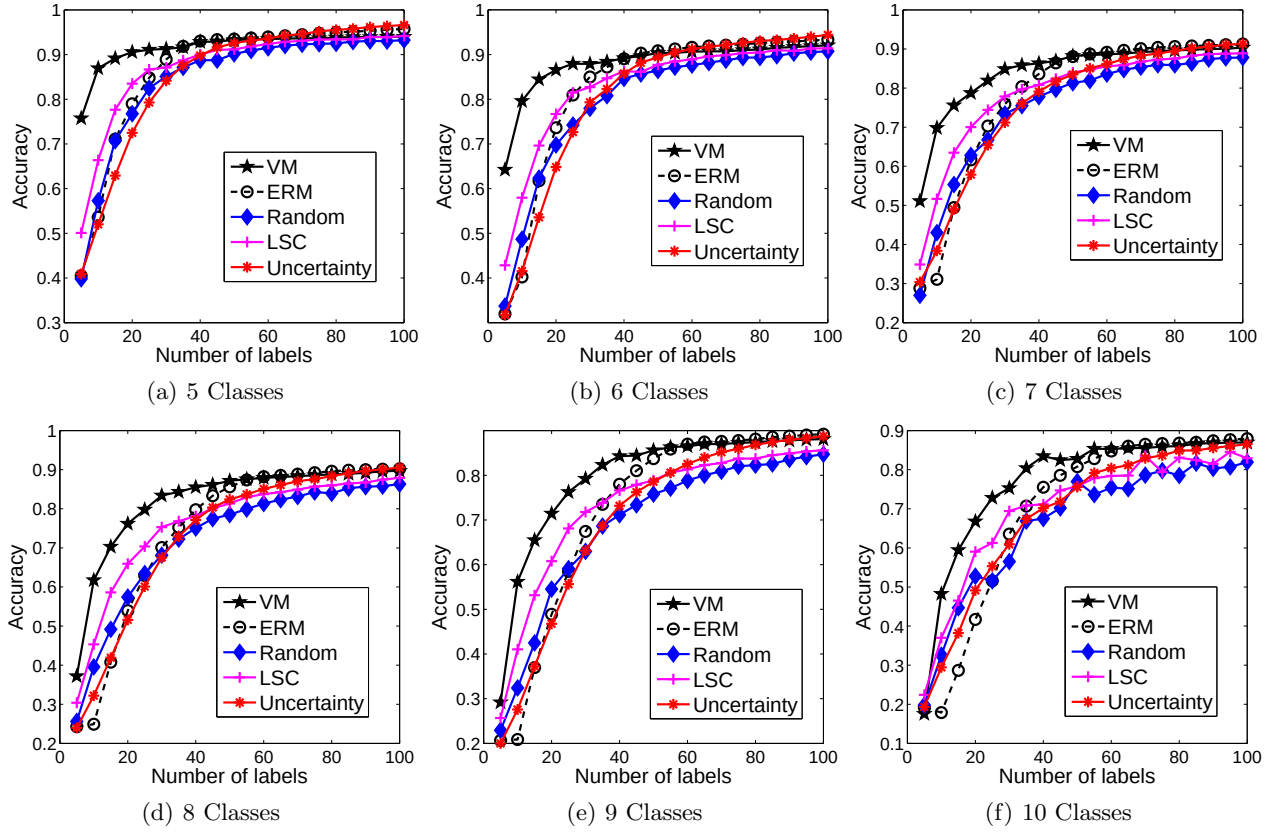
Three real-world data sets are used in our experiments. The first one is the Isolet spoken letter database [1]. It contains 150 subjects who spoke the name of each letter of the alphabet twice. Hence, we have 52 examples from each speaker. The speakers are grouped into sets of 30 speakers each, and are referred to as Isolet1, Isolet2, Isolet3, Isolet4, and Isolet5. Here we use Isolet1 which contains 1560 data instances of 26 classes (spoken letters). Each class has 60 examples, and each example is represented by a 617-dimensional vector recording the spectral coefficients, contour features, sonorant features, pre-sonorant features and post-sonorant features.

The second one is the MNIST handwritten digit

[1] http://archive.ics.uci.edu/ml/datasets/ISOLET

Table 2: Classification accuracy (%) by using 20 and 50 labels on the MNIST data set.

	5 classes		6 classes		7 classes		8 classes		9 classes		10 classes		average	
# of labels	20	50	20	50	20	50	20	50	20	50	20	50	20	50
VM	**90.6**	93.3	**86.6**	90.4	**78.8**	**88.2**	**76.2**	**87.2**	**71.4**	**85.6**	**66.8**	**82.8**	**78.4**	**87.9**
ERM	79.0	**93.5**	73.6	**90.8**	61.6	88.0	54.0	85.6	48.9	83.7	41.7	80.8	59.8	87.1
Random	76.8	90.2	69.8	86.4	62.8	81.3	57.4	78.5	54.5	75.8	52.8	77.0	62.4	81.5
LSC	83.5	91.2	76.7	87.7	70.0	84.0	65.9	81.4	60.8	78.9	59.0	75.8	69.3	83.2
Uncertainty	72.5	92.6	64.8	89.4	57.9	83.5	51.6	82.4	46.8	78.6	49.1	75.6	57.1	83.7



(a) 5 Classes (b) 6 Classes (c) 7 Classes

(d) 8 Classes (e) 9 Classes (f) 10 Classes

Figure 2: Classification accuracy vs. the number of labels used on the MNIST data set

database [2]. This database has a training set of 60,000 images (denoted as set 1), and a testing set of 10,000 images (denoted as set 2). We take the first 1000 images from set 1 and the first 1000 images from set 2 as our experimental data. Each class (digit) contains around 200 images, each of which is of size 28×28 and therefore represented by a 784-dimensional vector.

The third data set is a connected co-author graph extracted from the DBLP database [3] on four areas: machine learning, data mining, information retrieval and database, which naturally form four classes. The co-author graph contains a total of 1711 vertices, each of which represents an author. The edge between each pair of authors is weighted by the number of papers they co-authored. Each class (research area) contains around 400 authors.

[2] http://yann.lecun.com/exdb/mnist/

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

For each of the first two data sets, Isolet and MNIST, following [9], we build a 4-nearest neighbor graph among the data points, and run the active learning algorithms on graphs as well as the harmonic Gaussian field classifier. The third data set contains an inherent graph structure. Note that each data instance (author) in the co-author graph does not have a natural feature representation, therefore existing feature-based active learning methods cannot be directly applied to it.

4.2 Classification Results

For the Isolet and MNIST data sets, the experiments are conducted by choosing different numbers of classes (denoted as k) from the original data set. For Isolet, $k = 10, 15, 20, 25, 26$. For each given class number $k(= 10, 15, 20, 25)$, the performance scores are computed by averaging the scores of 10 repeats of different randomly chosen classes. When $k = 26$, which is the

Table 3: Classification accuracy (%) by using 20 and 50 labels on the co-author graph.

# of labels	20	50
VM	**50.4**	**62.2**
ERM	47.0	54.7
Random	41.7	50.7
LSC	30.0	54.3
Uncertainty	39.4	54.1

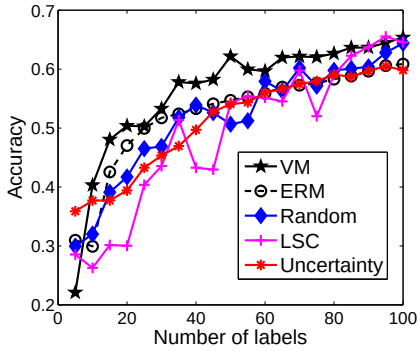


Figure 3: Classification accuracy vs. the number of labels used on the co-author graph.

total number of classes in Isolet, we report the performance scores of using the whole data set. For each test, we employ different active learning methods to select l examples to label and train a harmonic Gaussian classifier to predict the labels of the rest of the data. Fig. 1 shows the plots of classification accuracy versus the number of labels used (l). For MNIST, the number of classes is chosen to be $k = 5, 6, 7, 8, 9, 10$, and we also average the classification accuracy over 10 different random selections of classes except for $k = 10$, which corresponds to using the whole data set. The classification accuracy versus the number of labels used is plotted in Fig. 2. For the co-author graph, since the original data set only contains four classes, we directly run experiments on the whole data set. We show the performance comparison in Fig. 3.

As can be observed from Fig. 1 to Fig. 3, our proposed **VM** algorithm significantly outperforms other active learning criteria on all the three data sets, especially when the number of labels is very small. **LSC** performs the second best on the Isolet and MNIST data sets when the number of labels is relatively small. It is interesting to note that on the MNIST data set, **ERM** and **Uncertainty** perform not very well when the number of labels is small, and perform much better when more labels are selected, indicating that they rely heavily on the label information of the selected data.

We further provide the detailed classification accuracy by using 20 and 50 labels in Table 1∼3. The last two columns of Table 1 and Table 2 record the average classification accuracy over different numbers of classes.

We can see that overall, **VM** performs significantly better than all the other methods, including **ERM** and **Uncertainty** that use label information. Comparing with the algorithm that performs the second best in each case, **VM** achieves 27.0% (10.6%), 29.6% (6.2%), 6.4% (16.6%) relative error reduction in the average classification accuracy using 20 (50) labels on Isolet, MNIST and the co-author graph, respectively. We have also performed the two-tailed t-tests at 95% significance level over the experimental results in Table 1∼3. In all the cases that **VM** performs the best, the p-values between the results of **VM** and other algorithms are less than 0.05. Therefore, the improvements of our proposed algorithm are statistically significant.

5 Conclusions

From the variance minimization perspective, this paper proposes a novel active learning algorithm purely based on the graph structure, without label information and feature representation on the nodes. One key advantage over existing methods is that our method theoretically minimizes the expected prediction error of a popular graph-based classifier in a batch, offline mode. Experiments validate the effectiveness of our approach compared to existing active learners on graphs.

This study is based on the harmonic Gaussian field classifier. There are many other effective graph-based classifiers with different statistical assumptions of the distribution of the graph data. Therefore, it is worthwhile to further analyze the variance and expected prediction error of other learning models to guide the label selection over graphs. Moreover, this paper selects data to label purely based on the graph structure. In the future, when the feature representation of the nodes is also available, it will be interesting to combine the feature-based active learning criterion and the graph-based active learner together to select data.

6 Acknowledgements

The work was supported in part by U.S. National Science Foundation grants IIS-0905215, IIS-1017362, and the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NS-CTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] Y. Bengio, O. Delalleau, and N. Le Roux. Label propagation and quadratic criterion. In *Semi-Supervised Learning*, pages 193–216. MIT Press, 2006.

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

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

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, 1998.

[5] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[6] F. R. K. Chung. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics*. AMS, 1997.

[7] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[8] G. H. Golub and C. F. V. Loan. *Matrix computations*. Johns Hopkins University Press, 3rd edition, 1996.

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

[10] S. Hanneke. An analysis of graph cut size for transductive learning. In *ICML*, pages 393–399, 2006.

[11] D. A. Harville. *Matrix Algebra from a Statistician's Perspective*. Springer-Verlag, 1997.

[12] X. He. Laplacian Regularized D-Optimal Design for Active Learning and Its Application to Image Retrieval. *IEEE Transactions on Image Processing*, 19(1):254–263, 2010.

[13] X. He, M. Ji, C. Zhang, and H. Bao. A variance minimization criterion to feature selection using laplacian regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):2026–2038, 2011.

[14] S.-J. Huang, R. Jin, and Z.-H. Zhou. Active learning by querying informative and representative examples. In *NIPS*, pages 892–900, 2010.

[15] A. Kuwadekar and J. Neville. Combining semi-supervised learning and relational resampling for active learning in network domains. In *the Budgeted Learning Workshop, ICML*, 2010.

[16] J. Long, J. Yin, W. Zhao, and E. Zhu. Graph-based active learning based on label propagation. In *MDAI*, pages 179–190, 2008.

[17] S. A. Macskassy. Using graph-based metrics with empirical risk minimization to speed up active learning on networked data. In *KDD*, pages 597–606, 2009.

[18] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

[19] K. Pelckmans, J. Shawe-Taylor, J. A. K. Suykens, and B. D. Moor. Margin based transductive graph cuts using linear programming. *Journal of Machine Learning Research - Proceedings Track*, 2:363–370, 2007.

[20] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2010.

[21] L. Shi, Y. Zhao, and J. Tang. Combining link and content for collective active learning. In *CIKM*, pages 1829–1832, 2010.

[22] Z.-J. Zha, M. Wang, Y.-T. Zheng, Y. Yang, R. Hong, and T.-S. Chua. Interactive video indexing with statistical active learning. *IEEE Transactions on Multimedia*, 14(1):17–27, 2012.

[23] W. Zhao, J. Long, E. Zhu, and Y. Liu. A scalable algorithm for graph-based active learning. In *Frontiers in Algorithmics*, pages 311–322, 2008.

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

[25] X. Zhu, J. Lafferty, and Z. Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *the workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, ICML*, pages 58–65, 2003.