GRAPH-BASED ACTIVE LEARNING: A NEW LOOK AT EXPECTED ERROR MINIMIZATION

Kwang-Sung Jun and Robert Nowak

Wisconsin Institutes for Discovery, University of Wisconsin-Madison
kjun@discovery.wisc.edu, rdnowak@wisc.edu

ABSTRACT

In graph-based active learning, algorithms based on expected error minimization (EEM) have been popular and yield good empirical performance. The exact computation of EEM optimally balances exploration and exploitation. In practice, however, EEM-based algorithms employ various approximations due to the computational hardness of exact EEM. This can result in a lack of either exploration or exploitation, which can negatively impact the effectiveness of active learning. We propose a new algorithm TSA (Two-Step Approximation) that balances between exploration and exploitation efficiently while enjoying the same computational complexity as existing approximations. Finally, we empirically show the value of balancing between exploration and exploitation in both toy and real-world datasets where our method outperforms several state-of-the-art methods.

Index Terms— Machine learning, active learning, semi-supervised learning, graph-based learning, probabilistic model

1. INTRODUCTION

[1]This paper studies the problem of the graph-based active learning. We are given a weighted undirected graph $G = (N, E)$ with nodes $N = \{1, \dots, n\}$, edges E, and weights $w_{ij} = w_{ji} \geq 0, \forall i \leq j$, that are 0 if there is no edge between i and j. Each node $i \in N$ has a label $Y_i \in \{1, -1\}$[2]. Let $\boldsymbol{\ell}_1 \subseteq N$ be the initial labeled nodes. Initially, an algorithm knows the labels of $\boldsymbol{\ell}_1$ only. At each time step $t = 1, 2, \dots$, an algorithm must perform

1. PREDICT: Make label prediction $\widehat{Y}_i$ on each unlabeled nodes $i \notin \boldsymbol{\ell}_t$. Let $\widehat{Y}_i := Y_i, \forall i \in \boldsymbol{\ell}_t$. An algorithm suffers error rate $\epsilon_t = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{\widehat{Y}_i \neq Y_i\}$, which is *unknown* to the algorithm.
2. QUERY: Select an unlabeled node q and query its label. Receive the label Y_q. Update $\boldsymbol{\ell}_{t+1} = \boldsymbol{\ell}_t \cup \{q\}$.

The goal is to achieve a low error rate while querying as few nodes as possible. The problem PREDICT is an instance of semi-supervised learning [2] for which the seminal work of Zhu et al. [3] has been successful and de facto standard, which we call label propagation (**LP**). We thus focus on QUERY.

[1]A longer version of this paper is available on arXiv [1].

[2]A multi-class generalization is straightforward via the one-vs-the-rest reduction; see Section 4 for detail.

There are many examples where the data is given by or constructed as a graph. In document classification problems, two documents tend to be of the same topic when one cites the other or when they use the same keywords. A graph can be constructed based on such relations. The graph can then be used to infer a given document's topic from the known topics of the other connected documents. More generally, a graph can be constructed based on known similarities or dissimilarities between unlabeled examples in any machine learning application. For example, hand-written digits can be recognized efficiently through graph-based learning algorithms [3]. In all these examples, the edge weights in the graph carries important information on how strongly two nodes (examples) are related, which can be used to make label predictions.

One popular approach to QUERY starts from an intuitive probabilistic model. Consider the following probabilistic model for the random variable $\mathbf{Y} \in \{1, -1\}^n$:

$$\mathbb{P}(\mathbf{Y} = \mathbf{y}) = \frac{1}{Z} \exp\left(-\frac{\beta}{2}\sum_{i<j} w_{ij}(y_i - y_j)^2\right), \quad (1)$$

where Z is the normalization factor and $\beta > 0$ is a strength parameter. The model prefers labelings $\mathbf{y} \in \{1, -1\}^n$ that vary smoothly across edges; i.e., larger weight w_{ij} implies higher likelihood of $y_i = y_j$. We refer to the model above as binary Markov random field (**BMRF**). Note that BMRF would be equivalent to the Gaussian random field (GRF) if we relax the labels to belong to real values: $\mathbf{Y} \in \mathbb{R}^n$.

If the labels $\mathbf{Y}$ truly follow BMRF with known β, given a set of observed labels of nodes $\boldsymbol{\ell} \subseteq N$, the expected error rate of a prediction strategy is well-defined; e.g., see (4). Then, querying the node that minimizes the expected error in the next time step is a reasonable greedy strategy. We refer to the query strategy above as *expected error minimization* (**EEM**) principle. We precisely define EEM in Section 2.

EEM has been the main idea of many studies [4, 5, 6]. Define $\mathbf{Y}_{\boldsymbol{\ell}} := \{\mathbf{Y}_i\}_{i \in \boldsymbol{\ell}}$. The challenge in EEM is to compute the posterior marginal of a node i given labeled nodes $\boldsymbol{\ell} \subseteq N$:

$$\mathbb{P}(Y_i \mid \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}}). \quad (2)$$

This is combinatorial; there is no known polynomial time algorithm for computing it, to our knowledge. Resolving

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Error rate
True label	+	+	+	+	+	+	+	+	+	-	-	-	+	+	+	+	+	+	
ZLG	✓	+	+	+	+	✓	+	✓	✓	✓	✓	-	-	-	-	-	-	-	0.33
SOpt	✓	+	✓	+	+	✓	+	+	-	-	✓	-	✓	+	+	✓	+	+	0.06
BMRF	✓	+	+	+	+	✓	+	✓	+	-	✓	-	✓	+	+	✓	+	+	0.00
TSA (**Ours**)	✓	+	+	+	+	✓	+	✓	+	-	✓	-	✓	+	+	✓	+	+	0.00

Fig. 1. A linear chain example that contrasts different QUERY Algorithms

such a computational issue in EEM has been an active area of research. Zhu et al. [4] apply a simple approximation to (2) by posterior mean of GRF, which we call **ZLG**. V-optimality (**VOpt**) [5] considers EEM under GRF instead of BMRF, which results in a closed-form solution. Σ-optimality (**SOpt**) [6] takes the same approach as VOpt, but based on a different error notion called survey error.

Each EEM-based algorithm has an undesirable behavior. Consider a linear chain of length 18 with edges between i and $i+1$ for all $1 \le i \le 17$ with weight 1; see Figure 1. Labels for node 1 and 11 are given as initial labels. We denote labeled nodes by ✓ where initial labels are in gray, the first two queries are in black, and the last two are in red. Symbols +/- indicate the predicted labels by LP after 4 queries. For the first query, an algorithm sees that there is at least one cut (edge connecting different labels) between node 1 and 11. ZLG drills into this region and spends its next four queries in nailing down the cut. Consequently, it does not query any node to the right side of node 11 and incurs large error; i.e., ZLG lacks exploration queries. In SOpt, the first two queries does include exploration query (node 16). Then, the next two queries include node 3 that does not reduce the error rate; node 8 would have reduced error. SOpt selects queries by which *nodes* have been labeled, ignoring what *labels* they have. In fact, this is the common characteristic of many graph-based active learning algorithms [7, 8, 9]. This is why SOpt is not able to optimize exploitation queries, which results in higher error than other methods as we show in toy experiments in Section 4. VOpt shares the same issue, so we omit it here. In contrast, the exact computation of EEM (row BMRF) balances between exploration and exploitation.

In this work, we propose a new algorithm **TSA** whose name comes from a two-step approximation to the posterior marginal (2). TSA improves upon both ZLG and SOpt without added computational complexity. The time complexity of TSA per query is $O(n^2)$, which is the same as ZLG and SOpt. Unlike ZLG and SOpt, TSA balances between exploration and exploitation. In a linear chain example in Figure 1, TSA finds the same queries as BMRF. We present TSA in Section 3 and empirical results in Section 4 where we observe that TSA outperforms baseline methods on several toy and real-world datasets.

2. EXPECTED ERROR MINIMIZATION (EEM)

Consider a probabilistic model over a $\mathbf{Y} \in \{1,-1\}^n$ such as (1). Given a set of labeled nodes $\boldsymbol{\ell} \subseteq N$ with label $\mathbf{y}_{\boldsymbol{\ell}}$, the optimal prediction is the Bayes decision rule

$$\widehat{Y}_i(\mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}}) := \arg\max_{y \in \{1,-1\}} \mathbb{P}(Y_i = y \mid \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}}) \ . \quad (3)$$

Note $\widehat{Y}_i(\mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}}) = Y_i$ for $i \in \boldsymbol{\ell}$ trivially. We hereafter use $\widehat{Y}_i$ and omit $(\mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}})$ when it is clear from the context.

Define the unlabeled nodes $\mathbf{u} := N \backslash \boldsymbol{\ell}$. We are interested in measuring the expected error rate of the Bayes decision rule after querying $q \in \mathbf{u}$. Since we do not know Y_q yet, we take expectation over $Y_q \in \{1,-1\}$ as well as $\{Y_i\}_{i \in \mathbf{u} \backslash \{q\}}$. We define the expected error after knowing the label Y_q as follows, which we call *lookahead zero-one risk* of node q:

$$R^{+q}(\mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}}) := \mathbb{E}_{Y_q} \mathbb{E}_{\mathbf{Y}_{\mathbf{u} \backslash \{q\}}} \left[\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{\widehat{Y}_i \neq Y_i\} \,\middle|\, Y_q, \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}} \right] , \quad (4)$$

where $\widehat{Y}_i$ depends on Y_q as well as $\mathbf{Y}_{\boldsymbol{\ell}}$. We use $R^{+q}(\mathbf{y}_{\boldsymbol{\ell}})$ as a shortcut for $R^{+q}(\mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}})$.

The expected error minimization (**EEM**) principle is to choose the query that minimizes the lookahead zero-one risk:

$$\arg\min_{q \in N \backslash \boldsymbol{\ell}} R^{+q}(\mathbf{y}_{\boldsymbol{\ell}}) \ . \quad (5)$$

Define $\mathbb{P}_{\mathbf{y}_{\boldsymbol{\ell}}}(\cdot) := \mathbb{P}(\cdot | \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}})$ and the *zero-one risk*

$$\begin{aligned} & R(Y_q = y, \mathbf{y}_{\boldsymbol{\ell}}) \\ & := \mathbb{E}_{\mathbf{Y}_{\mathbf{u} \backslash \{q\}}} \left[\sum_{i=1}^{n} \frac{1}{n} \mathbb{1}\{\widehat{Y}_i \neq Y_i\} \,\middle|\, Y_q = y, \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}} \right] \\ & = \frac{1}{n} \sum_{i=1}^{n} \left(1 - \max_{y' \in \{1,-1\}} \mathbb{P}_{Y_q = y, \mathbf{y}_{\boldsymbol{\ell}}}(Y_i = y') \right) \ . \end{aligned} \quad (6)$$

Then,

$$R^{+q}(\mathbf{y}_{\boldsymbol{\ell}}) = \sum_{y \in \{1,-1\}} R(Y_q = y, \mathbf{y}_{\boldsymbol{\ell}}) \mathbb{P}_{\mathbf{y}_{\boldsymbol{\ell}}}(Y_q = y) \ . \quad (7)$$

Notice that the key quantity is the posterior marginal distribution $\mathbb{P}_{Y_q = y, \mathbf{y}_{\boldsymbol{\ell}}}(Y_i = y')$ in computing (6) and $\mathbb{P}_{\mathbf{y}_{\boldsymbol{\ell}}}(Y_q = y)$ in (7). An efficient computation of the posterior marginal would lead to an algorithm for PREDICT due to (3), and also to an algorithm for QUERY due to (5).

3. TWO-STEP APPROXIMATION OF MARGINAL

Consider BMRF defined in (1). Let $\mathbf{L}$ be the graph Laplacian defined by $L_{ij} := \mathbb{1}\{i = j\}(\sum_{k=1}^{n} w_{ik}) - w_{ij}$. We rewrite (1) compactly:

$$\mathbb{P}(\mathbf{Y} = \mathbf{y}) = \frac{1}{Z} \exp\left(-\frac{\beta}{2} \mathbf{y}^\top \mathbf{L} \mathbf{y} \right) \ . \quad (8)$$

For ease of exposition, we let $\beta = 1$; one can obtain results for $\beta \neq 1$ by replacing $\mathbf{L}$ with $\beta\mathbf{L}$.

Suppose we have observed the labels of nodes $\boldsymbol{\ell}$ as $\mathbf{y}_{\boldsymbol{\ell}}$. We propose a *two-step approximation* (**TSA**) to the posterior marginal distribution $\mathbb{P}_{\mathbf{y}_{\boldsymbol{\ell}}}(Y_k)$, which leads to a new QUERY algorithm. The key lies in the following log probability ratio approximation: $\log \frac{\mathbb{P}(Y_k=1,\mathbf{Y}_{\boldsymbol{\ell}}=\mathbf{y}_{\boldsymbol{\ell}})}{\mathbb{P}(Y_k=-1,\mathbf{Y}_{\boldsymbol{\ell}}=\mathbf{y}_{\boldsymbol{\ell}})} \approx \log \frac{\mu(Y_k=1,\mathbf{Y}_{\boldsymbol{\ell}}=\mathbf{y}_{\boldsymbol{\ell}})}{\mu(Y_k=-1,\mathbf{Y}_{\boldsymbol{\ell}}=\mathbf{y}_{\boldsymbol{\ell}})}$ for some $\mu(\cdot)$. Define the sigmoid function $\sigma(z) := (1 + \exp(-z))^{-1}$. Then, it follows that

$$\begin{aligned}
&\mathbb{P}(Y_k = 1 \mid \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}}) \\
&= \frac{\mathbb{P}(Y_k = 1, \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}})}{\mathbb{P}(Y_k = 1, \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}}) + \mathbb{P}(Y_k = -1, \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}})} \\
&= \sigma(\log \mathbb{P}(Y_k = 1, \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}}) - \log \mathbb{P}(Y_k = -1, \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}})) \\
&\approx \sigma(\log \mu(Y_k = 1, \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}}) - \log \mu(Y_k = -1, \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}})) \,.
\end{aligned}$$

We construct $\mu(Y_k = y_k, \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}})$ as a two-step upper-bound on $\mathbb{P}(Y_k = y_k, \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}})$. Define $\overline{\mathbf{u}} := \mathbf{u}\backslash\{k\}$, the set of unlabeled nodes except k. Let $A := \mathbf{L}_{kk} + \mathbf{y}_{\boldsymbol{\ell}}^\top \mathbf{L}_{\boldsymbol{\ell}\boldsymbol{\ell}}\mathbf{y}_{\boldsymbol{\ell}}$ and $g(\mathbf{y}_{\overline{\mathbf{u}}}) := -\big(\frac{1}{2}\mathbf{y}_{\overline{\mathbf{u}}}^\top \mathbf{L}_{\overline{\mathbf{u}}\overline{\mathbf{u}}}\mathbf{y}_{\overline{\mathbf{u}}} + y_k \mathbf{L}_{k\overline{\mathbf{u}}}\mathbf{y}_{\overline{\mathbf{u}}} + \mathbf{y}_{\boldsymbol{\ell}}^\top \mathbf{L}_{\boldsymbol{\ell}\overline{\mathbf{u}}}\mathbf{y}_{\overline{\mathbf{u}}}\big)$. We simplify $\log \mathbb{P}(Y_k = y_k, \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}}) =$

$$-\log(Z) - \frac{1}{2}A - y_k \mathbf{L}_{k\boldsymbol{\ell}}\mathbf{y}_{\boldsymbol{\ell}} + \log\Big(\sum_{\mathbf{y}_{\overline{\mathbf{u}}}} \exp(g(\mathbf{y}_{\overline{\mathbf{u}}}))\Big) \,.$$

Note that the last term is the log-sum-exp function that is similar to the max operator. This leads to our **first upperbound**:

$$\log\left(\sum_{\mathbf{y}_{\overline{\mathbf{u}}}} \exp(g(\mathbf{y}_{\overline{\mathbf{u}}}))\right) \leq \max_{\mathbf{y}_{\overline{\mathbf{u}}} \in \{1,-1\}^{|\overline{\mathbf{u}}|}} g(\mathbf{y}_{\overline{\mathbf{u}}}) + |\overline{\mathbf{u}}| \log 2.$$

We now have an integer optimization problem, which is hard in general. We relax the domain of $\mathbf{y}_{\overline{\mathbf{u}}}$ to real, which leads to our **second upperbound**:

$$\max_{\mathbf{y}_{\overline{\mathbf{u}}} \in \{1,-1\}^{|\overline{\mathbf{u}}|}} g(\mathbf{y}_{\overline{\mathbf{u}}}) \leq \max_{\mathbf{y}_{\overline{\mathbf{u}}} \in \mathbb{R}^{|\overline{\mathbf{u}}|}} g(\mathbf{y}_{\overline{\mathbf{u}}}).$$

We now have a concave quadratic maximization problem. Find the closed form solution. Then, altogether,

$$\begin{aligned}
&\log \mathbb{P}(Y_k = y_k, \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}}) \\
&\quad \leq -\log(Z) - \frac{1}{2}A - y_k \mathbf{L}_{k\boldsymbol{\ell}}\mathbf{y}_{\boldsymbol{\ell}} + \\
&\qquad \frac{1}{2}(y_k \mathbf{L}_{k\overline{\mathbf{u}}} + \mathbf{y}_{\boldsymbol{\ell}}^\top \mathbf{L}_{\boldsymbol{\ell}\overline{\mathbf{u}}})\mathbf{L}_{\overline{\mathbf{u}}\overline{\mathbf{u}}}^{-1}(\mathbf{L}_{\overline{\mathbf{u}}k} y_k + \mathbf{L}_{\overline{\mathbf{u}}\boldsymbol{\ell}}\mathbf{y}_{\boldsymbol{\ell}}) + |\overline{\mathbf{u}}| \log 2 \\
&\quad =: \log \mu(Y_k = y_k, \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}}) \,,
\end{aligned}$$

where $\mathbf{L}_{\overline{\mathbf{u}}\overline{\mathbf{u}}}^{-1} = (\mathbf{L}_{\overline{\mathbf{u}}\overline{\mathbf{u}}})^{-1}$. Let $f_k := \log \mu(Y_k = 1, \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}}) - \log \mu(Y_k = -1, \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}})$ be the decision value of node k. We simplify f_k:

$$f_k = -2\mathbf{L}_{k\boldsymbol{\ell}}\mathbf{y}_{\boldsymbol{\ell}} + 2\mathbf{L}_{k\overline{\mathbf{u}}}\mathbf{L}_{\overline{\mathbf{u}}\overline{\mathbf{u}}}^{-1}\mathbf{L}_{\overline{\mathbf{u}}\boldsymbol{\ell}}\mathbf{y}_{\boldsymbol{\ell}} \,. \tag{9}$$

for which we present a natural interpretation in our arXiv paper [1]. Finally, compute $\mathbb{P}(Y_k = 1 \mid \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}}) \approx \sigma(f_k)$ for all $k \notin \boldsymbol{\ell}$ and perform EEM (5).

Computing Marginals Altogether Note that we need to compute $\mathbf{p}(Y_k = 1 \mid \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}})$ for every node $k \in \mathbf{u}$, and the matrix inversion in (9) is costly. Denote by $\circ$ the Hadamard product and $[f_k]_{k\in\mathbf{u}}$ a vector whose k-th component has value f_k. Note that the one-step covariance update rule says that $\begin{pmatrix} \mathbf{L}_{\overline{\mathbf{u}}\overline{\mathbf{u}}}^{-1} & 0 \\ 0 & 0 \end{pmatrix} = \mathbf{L}_{\mathbf{u}\mathbf{u}}^{-1} - \frac{(\mathbf{L}_{\mathbf{u}\mathbf{u}}^{-1})_{\cdot k}(\mathbf{L}_{\mathbf{u}\mathbf{u}}^{-1})_{k\cdot}}{(\mathbf{L}_{\mathbf{u}\mathbf{u}}^{-1})_{kk}}$, where we assume that node k is the largest index among $\overline{\mathbf{u}}$, without loss of generality. Using the one-step covariance update, one can compute the marginals altogether with one matrix inversion (see [1]):

$$[f_k]_{k\in\mathbf{u}} = -2\left[\frac{1}{(\mathbf{L}_{\mathbf{u}\mathbf{u}}^{-1})_{kk}}\right]_k \circ (\mathbf{L}_{\mathbf{u}\mathbf{u}}^{-1}\mathbf{L}_{\mathbf{u}\boldsymbol{\ell}}\mathbf{y}_{\boldsymbol{\ell}}) \,. \tag{10}$$

Evaluating (5) involves computing (10) n times. Since the matrix inversion in (10) can be performed in $O(n^2)$ using the one-step covariance update, the time complexity would be $O(n^3)$. However, one can use the "dongle node" trick presented in Appendix A of [4] to reduce it to $O(n^2)$; see [1].

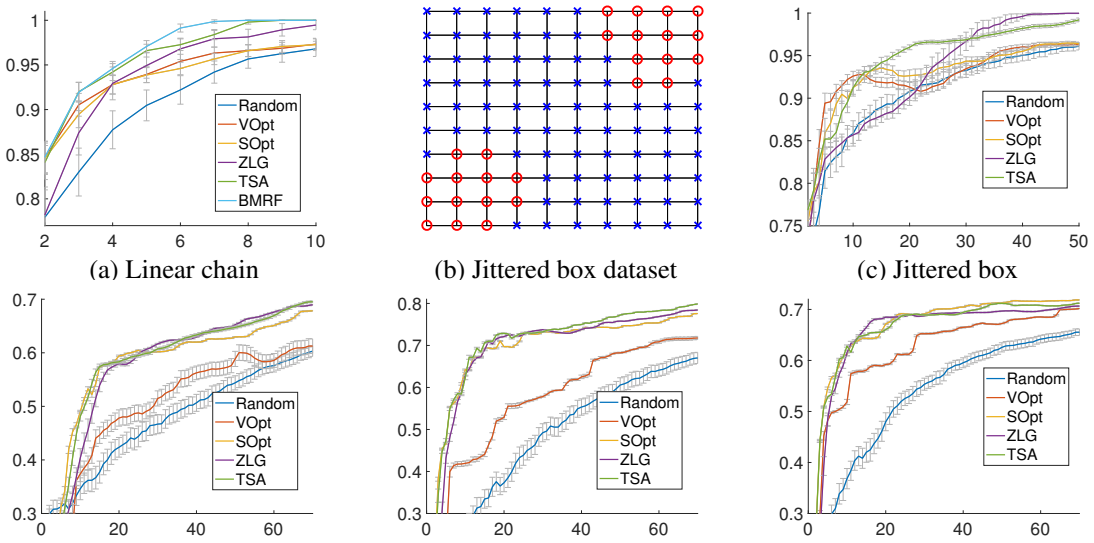
Comparison to ZLG Let $\sigma^{\text{Linear}}(z) := \frac{1}{2}(z + 1)$ that is valid over $z \in [-1, 1]$ only. ZLG performs a simple approximation: $[\mathbb{P}(Y_k = 1 \mid \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}})]_k \approx \sigma^{\text{Linear}}(-\mathbf{L}_{\mathbf{u}\mathbf{u}}^{-1}\mathbf{L}_{\mathbf{u}\boldsymbol{\ell}}\mathbf{y}_{\boldsymbol{\ell}})$, where we apply σ^{Linear} elementwise. Input to σ^{Linear} is always in $[-1, 1]$ due to the property of the harmonic function [3]. In TSA, $[\mathbb{P}(Y_k = 1 \mid \mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}})]_k \approx$

$$\sigma\left(2 \cdot \left[\frac{1}{(\mathbf{L}_{\mathbf{u}\mathbf{u}}^{-1})_{kk}}\right]_k \circ (-\mathbf{L}_{\mathbf{u}\mathbf{u}}^{-1}\mathbf{L}_{\mathbf{u}\boldsymbol{\ell}}\mathbf{y}_{\boldsymbol{\ell}})\right) \,.$$

Both methods utilize $\mathbf{h} := (-\mathbf{L}_{\mathbf{u}\mathbf{u}}^{-1}\mathbf{L}_{\mathbf{u}\boldsymbol{\ell}}\mathbf{y}_{\boldsymbol{\ell}})$, which is the decision value of LP that is thresholded at 0 to make predictions (and notice both methods lead to the same prediction). Beside using a different sigmoid function, TSA further weights h_k by $1/(\mathbf{L}_{\mathbf{u}\mathbf{u}}^{-1})_{kk}$ where $(\mathbf{L}_{\mathbf{u}\mathbf{u}}^{-1})_{kk}$ is always positive. $(\mathbf{L}_{\mathbf{u}\mathbf{u}}^{-1})_{kk}$ can be interpreted as the variance of node k in GRF context. The larger the variance of a node is, the closer its decision value to 0, and the closer the marginal probability to 1/2. Such a variance information is not utilized in ZLG.

A striking example is our introductory example in Figure 1. When the initial labels are given for node 1 and 11, the posterior marginal $\mathbb{P}(Y_k = -1|\mathbf{Y}_{\boldsymbol{\ell}} = \mathbf{y}_{\boldsymbol{\ell}})$ for $k = 12, \ldots, 18$ under BMRF is (0.88, 0.79, 0.72, 0.67, 0.63, 0.60, 0.57) and under TSA is (0.88, 0.73, 0.66, 0.62, 0.60, 0.58, 0.57). Among node 12 to 18, node 16 has the smallest lookahead risk under both methods. However, under ZLG the marginals are (1,1,. . .,1) for node 12 to 18, which results in all 0 lookahead risk. Similarly, after querying node 6 the segment from node 2 to 5 have marginals (0,0,. . .,0) and 0 lookahead risk values. This explains why ZLG lacks exploration queries.

Another difference is that replacing occurrences of $\mathbf{L}$ with $\beta\mathbf{L}$ in ZLG results in no contribution of β whereas in TSA there *exists* contribution of β; the smaller the β is the closer the marginal probabilities to 1/2. We observed that β changes the balance between exploration and exploitation. However, parameter tuning in active learning is hard in general; we leave it as a future work and use $\beta = 1$ in experiments.



(a) Linear chain (b) Jittered box dataset (c) Jittered box

(d) DBLP (e) CORA (f) CITESEER

Fig. 2. Experiment Results. Plots show accuracy vs. the number of queries. Error bars are in gray.

4. EXPERIMENTS

Throughout the experiments, all methods start from one labeled node that is chosen uniformly at random. For every method, we break ties uniformly at random. Let C be the number of classes. We handle multi-class case by instantiating one algorithm for each one-vs-the-rest (total C runs). After computing each one-vs-the-rest marginal (binary), we compute the multi-class marginal distribution (now multinomial) by normalizing the binary marginals. Finally, the multi-class zero-one risk is a trivial extension of (6) from which we compute the EEM query.

Toy Data The first toy dataset is a linear chain with 15 nodes where each edge has weight 1. We choose an edge uniformly at random and assign positive label on one side and negative on the other side. We repeat the experiment 50 times where we assign new labels before each trial. We plot the accuracy vs. the number of queries in Fig. 2(a) with the confidence bounds in gray. After 10 queries, we observe a group of methods that outperforms the rest. This group consists of methods that are equipped with exploitation queries and thus able to nail down the exact cut. The rest are *non-adaptive* methods who are blind to observed labels. This experiment confirms the importance of the exploitation queries.

The second toy dataset is the 10 by 10 grid graph; see Fig. 2(b). We assign positive labels to the 3 by 3 box at the bottom left and another one at the top right, and negative labels to the rest. Then, for each negative nodes adjacent to a positive node, we assign positive with probability 1/2 to make the boundary "jittered". We repeat the experiment 50 times where we assign new jittered labels before each trial. We show the result in Fig. 2(c). There is no absolute winner. For very early time period, both VOpt and SOpt perform slightly better than the rest since they explore only — rough locations of the two positive boxes are discovered fast. On the other hand, ZLG incurs very low accuracy in the first half for the following two reasons: (i) before discovering a positive node, every node has the same lookahead risk and ZLG resorts to tie-breaking uniformly at random and (ii) after discovering the first positive node, ZLG drills down the exact boundary of it while completely not knowing the existence of the other positive box. In the end, however, ZLG becomes the best since it does not waste queries on exploration. TSA, our method, balances between exploration and exploitation and perform well on average.

Name	$\|N\|$	$\|E\|$	The number of classes
DBLP	1711	2898	4
CORA	2485	5069	7
CITESEER	2109	3665	6

Table 1. Real-world dataset summary

Real-World Data We use exactly the same dataset as [6][3], which is summarized in Table 1. DBLP is a coauthorship network, and both CORA and CITESEER are citation networks; see [6] for detail. We repeat the experiment 50 times and plot the results in Fig. 2(d-f). Overall, SOpt is better than ZLG for earlier time period, but ZLG is better for later time period (except in CITESEER), which we believe is due to the fact that ZLG lacks exploration queries and SOpt lacks exploitation queries, respectively. In contrast, TSA is as good as SOpt for earlier time period and as good as or even better than ZLG for later time period in all three datasets as TSA is able to balance between exploration and exploitation.

Acknowledgements

This work was partially supported by the National Science Foundation grants CCF-1218189 and IIS-1447449 and by MURI grant ARMY W911NF-15-1-0479.

[3]The dataset can be download from `http://www.autonlab.org/autonweb/21763`

5. REFERENCES

[1] Kwang-Sung Jun and Robert Nowak, "Graph-Based Active Learning: A New Look at Expected Error Minimization," *arXiv:1609.00845*, 2016.

[2] Xiaojin Zhu, "Semi-Supervised Learning Literature Survey," Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[3] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty, "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2003, pp. 912–919.

[4] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani, "Combining Active Learning and Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," in *ICML workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003, pp. 58–65.

[5] Ming Ji and Jiawei Han, "A Variance Minimization Criterion to Active Learning on Graphs," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012, pp. 556–564.

[6] Yifei Ma, Roman Garnett, and Jeff Schneider, "Sigma-Optimality in Active Learning on Gaussian Random Fields," in *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[7] Quanquan Gu and Jiawei Han, "Towards active learning on graphs: An error bound minimization approach," in *Proceedings - IEEE International Conference on Data Mining (ICDM)*, 2012, pp. 882–887.

[8] Akshay Gadde, Aamir Anis, and Antonio Ortega, "Active Semi-supervised Learning Using Sampling Theory for Graph Signals," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 492–501.

[9] Andrew Guillory and Jeff A Bilmes, "Label Selection on Graphs," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 691–699. 2009.