

Online Spectral Learning on a Graph with Bandit Feedback

Quanquan Gu

Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544
qgu@princeton.edu

Jiawei Han

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
hanj@illinois.edu

Abstract—Online learning on a graph is appealing due to its efficiency. However, existing online learning algorithms on a graph are limited to binary classification. Moreover, they require accessing the full label information, where the label oracle needs to return the true class label after the learner makes classification of each node. In many application scenarios, we only have access to partial label information, where the label oracle will return a single bit indicating whether the prediction is correct or not, instead of the true class label. This is also known as *bandit feedback*. In this paper, to overcome the above limitations of existing online learning algorithms on a graph, we study online learning on a graph for multi-class node classification, in both the full information setting and the partial information setting. First, we present an online multi-class classification algorithm in the full information setting. It is based on function learning on a graph using the spectral information of the graph Laplacian. We show that it attains $O(cd\log T)$ regret bound, where T is the number of rounds in online learning, c is the number of classes, and d is the number of eigenvectors of the graph Laplacian used for learning. Second, we present an online multi-class classification algorithm with bandit feedback. We use upper-confidence bound technique to trade off the exploration and exploitation of label information. We show that it attains $O(c\sqrt{T}\log T)$ regret bound, which is only a \sqrt{T} factor worse than the proposed algorithm in the full information setting. Experiments on several benchmark graph datasets show that the proposed online multi-class classification algorithm beats the state-of-art baseline, and the proposed bandit algorithm is also much better than the bandit version of the baseline.

Keywords-Online Learning on a Graph; Spectral Learning; Bandit Feedback; Partial Information Setting; Regret Bound; Graph Cut Size

I. INTRODUCTION

Existing online node classification algorithms on a graph [12] [11] [10] [9] [8] are limited to binary classification¹ and require the access of the full label information. In other words, in each round of online learning, after prediction the class label of a node, the learner is provided with the true label of that node. This is called *full information* setting.

This work was done when the first author worked at UIUC.

¹Although many of them used multi-class datasets for experimental study, they simply applied their algorithms to a set of binary classification datasets that are generated from those multi-class datasets, and used the average results for evaluation. In other words, they did not study multi-class classification.

However, similar to standard (semi-)supervised learning for vector data [20] [19], labels are often expensive and time consuming to obtain in graph classification. Therefore, it is desirable to access a small amount of labels, or *partial label information*. In this study, we are particularly interested in online multi-class classification with *bandit feedback* [15] [18] [7]: in each round, the learner receives a node and predicts its class label. Then the learner receives a single bit of feedback of whether the predicted label is correct or not. The advantage of bandit feedback is two fold: (i) it is able to reduce the labeling cost; and (ii) the single bit feedback is often more reliable, especially in the high-throughput data processing regime, where the oracle such as human worker has very limited time to label each node. Note that in the case of binary node classification, the single bit feedback actually reveals the true node label². However, in the multi-class node classification with c classes, we can identify the true label of a node only if the single bit feedback is positive. We cannot identify the true label of a node when the single bit feedback is negative, because there are still $c - 1$ possibilities for the true class label. Therefore, online multi-class classification with bandit feedback is more challenging than online multi-class classification in the full information setting.

In this paper, based on the above observation, we aim at designing online learning algorithms on a graph for multi-class node classification, both in the full information setting, as well as in the partial information setting. Our proposed algorithms are developed based on a spectral learning on a graph [21] [14]. In details, we use the spectral information of the graph Laplacian [6], i.e., eigenvalues and eigenvectors of the graph Laplacian, to learn multiple nonparametric functions on a graph for multi-class classification. We first present an online multi-class learning algorithm, namely *Online Spectral Learning on a Graph* (OSLG). We show that it attains $O(cd\log T)$ regret bound, where $T \leq n$ is the number of rounds in online learning, n is the graph size, c is the number of classes, and d is the number of eigenvectors used. Furthermore, the difficulty of online multi-class classification is characterized by the cut size of the graph.

²This implies that it is only meaningful to consider bandit feedback in the multi-class classification regime.

Second, we present an online multi-class classification algorithm with partial bandit feedback, namely *Online Spectral Learning on a Graph with Bandit feedback* (OSLG Bandit). Following the idea of upper-confidence bound technique invented in multi-armed bandit problem [2] [1] [5], we use upper-confidence bound of the classifier to trade off the exploration and exploitation of label information. We also show that its regret bound is $O(cd\sqrt{T} \log T)$, which is only a \sqrt{T} factor worse than that of OSLG. This is reasonable because OSLG uses more information than OSLG Bandit. The difficulty of online multi-class classification on a graph with partial feedback is characterized by the \sqrt{T} factor, and is also characterized by the graph cut size. Experiments on benchmark graph datasets show that OSLG outperforms multi-class GPA [11] greatly. Furthermore, our experiments also confirm the effectiveness of the online algorithm with bandit feedback (OSLG Bandit), which achieves much better results than the bandit version of multi-class GPA.

A. Notation and Organization

Throughout this paper, we will use lower case letters to denote scalars, lower case bold letters to denote vectors (e.g., \mathbf{w}), upper case letters to denote the elements of a matrix or a set, and bold-face upper case letters to denote matrices (e.g., \mathbf{A}). $\mathbf{0}$ is a vector of all zeros with appropriate length, and $\mathbf{1}$ is a vector of all ones with appropriate length. \mathbf{I} is an identity matrix with an appropriate size. We use \mathbf{w}^\top denote the transpose of a vector \mathbf{w} , and \mathbf{A}^{-1} the inverse of a matrix \mathbf{A} . $\text{diag}(\sigma_1, \dots, \sigma_n)$ denotes a diagonal matrix with diagonal elements equal to σ_i 's. Furthermore, $\|\cdot\|$ denotes the ℓ_2 -norm of a vector, $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, and $\|\mathbf{x}\|_{\mathbf{A}}$ denotes the matrix-induced norm $\|\mathbf{x}\|_{\mathbf{A}} = \sqrt{\mathbf{x}^\top \mathbf{A} \mathbf{x}}$.

The remainder of this paper is organized as follows. In Section II, we present an online spectral learning algorithm for multi-class node classification, followed by its regret bound. In Section III, we devise an online spectral learning algorithm for multi-class node classification with bandit feedback, and analyze its regret bound. The experiments on benchmark graph datasets are demonstrated in Section IV. Finally, we present the conclusions in Section V.

II. ONLINE SPECTRAL LEARNING ON A GRAPH

In this section, we present an online learning algorithm on a graph for multi-class node classification in the full information setting. We design our algorithm from the perspective of spectral learning, which learns nonparametric function(s) on a graph over a basis set that consists of eigenvectors of the graph Laplacian.

A. Spectral Learning Based on Graph Laplacian

Given a graph $G = (V, E)$, where $v_i \in V$ is the i -th node of a graph, and $e_{ij} \in E$ is the link (edge) between i -th node and the j -th node. Each link e_{ij} is associated

with a weight S_{ij} , which reflects the strength of the link. $\mathbf{S} \in \mathbb{R}^{n \times n}$ is called adjacency matrix of the graph. For undirected graph, \mathbf{S} is a symmetric matrix, while for directed graph, \mathbf{S} is asymmetric. In the setting of multi-class node classification, some of the nodes in the graph are labeled, i.e., $y_i \in \{1, \dots, c\}$, while the remainder are unlabeled, i.e., $y_i = 0$. Our goal is to obtain a prediction about the labels of those unlabeled nodes. For the ease of derivation, we introduce a label vector $\mathbf{y}_k = [y_{1k}, \dots, y_{nk}]^\top \in \mathbb{R}^n$ for the k -the class, where $y_{ik} = 1$ if $y_i = k$, and $y_{ik} = -1$ otherwise.

Let $f_k : V \rightarrow \mathbb{R}$ be a nonparametric function for the k -th class defined on the nodes of a graph. For a undirected graph $G = (V, E)$, the smoothness of f_k can be measured as follows [17] :

$$\frac{1}{2} \sum_{i,j=1}^n (f_{ik} - f_{jk})^2 S_{ij} = \mathbf{f}_k^\top \mathbf{L} \mathbf{f}_k, \quad (1)$$

where $\mathbf{f}_k = [f_{1k}, \dots, f_{nk}]^\top \in \mathbb{R}^n$ is the predicted label vector, f_{ik} is the function value of f_k on the i -th node, i.e., $f_{ik} = f_k(v_i)$. $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is the combinatorial graph Laplacian [6], where \mathbf{D} is a diagonal matrix, which is also known as the degree matrix. The i -th diagonal entry of \mathbf{D} is defined as $D_{ii} = \sum_{j=1}^n S_{ij}$. Roughly speaking, f_k is smooth if f_{ik} and f_{jk} are similar for those node pairs with large S_{ij} . This is sometimes informally expressed by saying that f_k changes slowly over the graph.

Suppose the eigenvalue decomposition of \mathbf{L} is $\mathbf{L} = \sum_{i=1}^n \sigma_i \mathbf{v}_i \mathbf{v}_i^\top$, where $0 \leq \sigma_1 \leq \sigma_2 \leq \dots \leq \sigma_n$ are eigenvalues, and $\mathbf{v}_i \in \mathbb{R}^n$, $i = 1, \dots, n$ are the corresponding eigenvectors. One property of the graph Laplacian is that its smallest eigenvalue is 0 (i.e., $\sigma_1 = 0$), and the associated eigenvector is 1. Through our paper, we assume that the graph G is connected, though our results can be generalized to disconnected graphs with more involved arguments. So the algebraic multiplicity of the zero eigenvalue is 1 (i.e., $\sigma_2 > 0$). Since an eigenvector can be seen as a nonparametric function on the graph, the smoothness of an eigenvector \mathbf{v}_i is measured by $\mathbf{v}_i^\top \mathbf{L} \mathbf{v}_i = \sigma_i$ [21], which is exactly the eigenvalue corresponding to this eigenvector. Thus, eigenvectors with smaller eigenvalues are smoother.

Since we want f_k , $k = 1, \dots, c$ to be smooth with respect to the graph G , following [21] [14], we assume that \mathbf{f}_k can be expanded by the $d \ll n$ eigenvectors of \mathbf{L} which correspond to the d smallest eigenvalues,

$$\mathbf{f}_k = \sum_{i=2}^d \frac{1}{\sqrt{\sigma_i}} \mathbf{v}_i w_{i,k} = \mathbf{M}^\top \mathbf{w}_k, \quad k = 1, \dots, c, \quad (2)$$

where $\mathbf{M} = [1/\sqrt{\sigma_2} \mathbf{v}_2, \dots, 1/\sqrt{\sigma_d} \mathbf{v}_d]^\top \in \mathbb{R}^{d \times n}$, $\mathbf{w}_k = [w_{1,k}, \dots, w_{d,k}]^\top \in \mathbb{R}^d$. Note that we use $1/\sqrt{\sigma_i}$ to normalize the corresponding eigenvectors \mathbf{v}_i to encourage large coefficients correspond to the eigenvectors with the small eigenvalues and vice versa. Since $\sigma_1 = 0$, the eigenvector

$\mathbf{v}_1 = \mathbf{1}$ is omitted. This is slightly different from the original formulation [21] [14] which did not use the eigenvalues for normalization. It turns out that our formulation is more effective according to our empirical study. Intuitively speaking, we not only use the eigenvectors, but also the eigenvalues. In this sense, our algorithm utilizes more spectral information than previous studies.

B. Algorithm

Now we are ready to propose the online multi-class classification algorithm based on (2). Before that, let us state the formal problem setting of online multi-class classification on a graph in the full information setting. From now on, we assume $T \leq n$. Let $\mathbf{M} = [\mathbf{m}_1, \dots, \mathbf{m}_T]$, where $\mathbf{m}_i \in \mathbb{R}^d$ is the i -th column of \mathbf{M} . Online learning operates on a sequence of nodes. In round t , the algorithm receives an incoming node $\mathbf{m}_t \in \mathbb{R}^d$, and predicts its label $\hat{y}_t \in \{1, \dots, c\}$ by $\hat{y}_t = \arg \max_k (\mathbf{w}_k^\top \mathbf{m}_t)$. After the prediction, the true label $y_t \in \{1, \dots, c\}$ is revealed and the loss $\ell_t(\cdot)$ is evaluated. The goal of online learning is to minimize the cumulative prediction error over the entire graph.

We present the online algorithm in Algorithm 1.

Algorithm 1 Online Spectral Learning on a Graph (OSLG)

```

Input: Adjacency matrix  $\mathbf{S}$ , rank  $d$ , regularization parameter  $\mu$ 
Output:  $\mathbf{W}_T$ 
Compute  $\mathbf{L} = \mathbf{D} - \mathbf{S}$  and  $\mathbf{M}$  from  $\mathbf{L}$ 
Initialize:  $\mathbf{A}_0 = \mu\mathbf{I}$  and  $\mathbf{b}_{k,0} = \mathbf{0}$  for  $k = 1, \dots, c$ ,  $\mathbf{W}_1 = \mathbf{0}$ 
for  $t = 1$  to  $T$  do
    Receive  $\mathbf{m}_t \in \mathbb{R}^d$ 
    Predict  $\hat{y}_t = \arg \max_{1 \leq k \leq c} (\mathbf{w}_{k,t}^\top \mathbf{m}_t)$ 
    Receive the correct label  $y_t \in \{1, \dots, c\}$ 
    if  $\hat{y}_t \neq y_t$  then
         $\mathbf{A}_t = \mathbf{A}_{t-1} + \mathbf{m}_t \mathbf{m}_t^\top$ 
        for  $k = 1$  to  $c$  do
             $\mathbf{b}_{k,t} = \mathbf{b}_{k,t-1} + y_{tk} \mathbf{m}_t$ 
             $\mathbf{w}_{k,t+1} = \mathbf{A}_{k,t}^{-1} \mathbf{b}_{k,t}$ 
        end for
    else
         $\mathbf{A}_t = \mathbf{A}_{t-1}$ 
        for  $k = 1$  to  $c$  do
             $\mathbf{b}_{k,t} = \mathbf{b}_{k,t-1}$ 
             $\mathbf{w}_{k,t+1} = \mathbf{w}_{k,t}$ 
        end for
    end if
end for

```

Note that in each iteration of Algorithm 1, whenever an update is invoked, the time complexity is $O(d^2)$. Since $d \ll n$, the computational time is acceptable.

C. Theoretical Analysis

Now we will prove the regret bound of Algorithm 1. This bound shows that, for any ordering of nodes on a graph, our algorithm cannot perform much worse than the best predictor learned in hindsight. We adapted the proof technique used in [4] [16] [3].

Define the loss function $\ell_t(\mathbf{W}) = \frac{1}{2} \sum_{k=1}^c (\mathbf{w}_k^\top \mathbf{m}_t - y_{tk})^2$. Then we can define the regret of OSLG as follows:

$$R_T = \sum_{t=1}^T \ell_t(\mathbf{W}_t) - \sum_{t=1}^T \ell_t(\mathbf{U}),$$

where $\mathbf{W}_t = [\mathbf{w}_{1,t}, \dots, \mathbf{w}_{c,t}]$ and $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_c]$.

For the ease of presentation, we define a set $\mathcal{M} = \{t : \hat{y}_t \neq y_t\}$, which is the set of nodes the algorithm makes mistake in prediction.

Theorem 1 (Regret Bound of OSLG) *Let $S = \{(\mathbf{m}_1, y_1), \dots, (\mathbf{m}_T, y_T)\} \in (\mathbb{R}^d \times \{1, \dots, c\})^T$. Then for any $\mathbf{U} \in \mathbb{R}^{d \times c}$ such that $(\mathbf{u}_k^\top \mathbf{m}_t - y_{tk})^2 \leq \gamma$, $\mathbf{F} \in \{-1, 1\}^{T \times c}$, Algorithm 1 has*

$$R_T \leq \mu \Phi(\mathbf{F}) + \gamma c d \log \left(1 + \frac{|\mathcal{M}| B^2}{d \mu} \right).$$

The regret bound of Algorithm 1 is $O(cd \log |\mathcal{M}|)$, where $|\mathcal{M}|$ is the number of mistakes the algorithm made. Since $|\mathcal{M}| \leq T$, we will simply use a slightly loose bound $O(cd \log T)$ to characterize the regret of OSLG in the rest of this paper.

On the other hand, there is a constant term proportional to $\Phi(\mathbf{F})$ in the regret bound. Note that $\Phi(\mathbf{F})$ is the graph cut size of a graph with respect to the partition denoted by \mathbf{F} . It indicates that the easier a graph can be partitioned (or clustered), the smaller the regret of Algorithm 1 will be. In other words, the (minimum) graph cut size characterizes the difficulty of online learning on a graph.

III. ONLINE SPECTRAL LEARNING ON A GRAPH WITH BANDIT FEEDBACK

In last section, we have studied online learning on a graph for multi-class node classification in the full information setting. In this section, we will present an online learning algorithm on a graph for multi-class node classification in the partial setting with bandit feedback. First of all, we formally give the definition of online learning on a graph with bandit feedback.

A. Algorithm

The generic problem of online multi-class classification over a graph in the bandit setting is as follows: in the t -th round of online learning, the learner receives a node \mathbf{m}_t to be classified, and predicts its label as $\hat{y}_t \in \{1, \dots, c\}$. After the prediction, the learner does not receive the correct label y_t but only a single bit M_t indicating whether its output \hat{y}_t is correct or not, i.e., $M_t = \{y_t \neq \hat{y}_t\}$.

In our paper, following [7], we assume that

$$P(Y_t = k | \mathbf{m}_t) = \frac{1 + \mathbf{u}_k^\top \mathbf{m}_t}{2},$$

for some $\mathbf{u}_k \in \mathbb{R}^d, k = 1, \dots, c$, subject to

$$\sum_{k=1}^c \mathbf{u}_k^\top \mathbf{m}_t = 2 - c, \quad -1 \leq \mathbf{u}_k^\top \mathbf{m}_t \leq 1.$$

Here \mathbf{u}_k is the Bayes classifier of unknown norm $\|\mathbf{u}_k\|$ which satisfies $|\mathbf{u}_k^\top \mathbf{m}_t| \leq 1$ for all t . We also define $\Delta_{k,t} = \mathbf{u}_k^\top \mathbf{m}_t$. We further define $\hat{\Delta}_{k,t} = \mathbf{w}_{k,t}^\top \mathbf{m}_t$, which is an estimator of $\Delta_{k,t}$.

We show the proposed online multi-class classification algorithm on a graph with bandit feedback in Algorithm 2. The algorithm we present employs linear models. The algorithm maintains c weight vectors $\mathbf{w}_k \in \mathbb{R}^d, k = 1, \dots, c$. Given a node \mathbf{m}_t , the algorithm computes a score associated with each of the c classes, denoted by $\mathbf{w}_k^\top \mathbf{m}_t$, and outputs a prediction to be the label with the highest score, i.e.,

$$\tilde{y}_t = \arg \min_k \mathbf{w}_k^\top \mathbf{m}_t.$$

We emphasize that two quantities are considered: the label with maximal score \tilde{y}_t , and the label that is actually output by the algorithm, denoted by \hat{y}_t . In the full information setting, most algorithms just output $\hat{y}_t = \tilde{y}_t$.

Since we only receive bandit feedback, there is a natural tradeoff between exploration and exploitation. On the one hand, the algorithm should output the label with highest scoring $\tilde{y}_t = \arg \max_k \mathbf{w}_k^\top \mathbf{m}_t$, which is called exploitation. On the other hand, the algorithm should perform exploration and output another label to get useful feedback. To achieve both exploration and exploitation, we propose to maintain the confidence information about the prediction. Specifically, given a node \mathbf{m}_t , the algorithm computes not only scores, but also the uncertainties for these scores, denoted by $\epsilon_{k,t}$. Intuitively, big values of $\epsilon_{k,t}$ indicate that the algorithm is less confident in its score $\mathbf{w}_k^\top \mathbf{m}_t$. Given a new node, the algorithm outputs the label with the highest upper confidence bound (UCB), computed as the sum of the score and its uncertainty

$$\hat{y}_t = \arg \max_k \mathbf{w}_k^\top \mathbf{m}_t + \epsilon_{k,t},$$

where $\epsilon_{k,t}$ is defined as

$$\epsilon_{k,t}^2 = \eta_t \mathbf{m}_t^\top \mathbf{A}_{k,t-1}^{-1} \mathbf{m}_t,$$

for some scalar η_t , which is used to trade-off the exploration and exploitation. This upper confidence bound technique has been widely used in multi-armed bandits problem [1] [2] [5].

Note that our algorithm shares similar spirit with confidence based bandit algorithm (Confidit) algorithm [7]. However, Confidit algorithm is designed for vector-based online learning. In addition, Confidit is a randomized algorithm while our algorithm is deterministic. More importantly, our

algorithm is projection-free. In other words, we do not need to do projection in each round of OSLG Bandit. This makes our algorithm computationally more efficient, without significantly affecting the theoretical properties.

Algorithm 2 Online Spectral Learning on a Graph with Bandit Feedback (OSLG Bandit)

Input: Adjacency matrix \mathbf{S} , rank d , regularization parameter μ and η_t .
Output: \mathbf{W}_T
Compute $\mathbf{L} = \mathbf{D} - \mathbf{S}$ and \mathbf{M} from \mathbf{L}
Initialize: $\mathbf{A}_{k,0} = \mu \mathbf{I}$ and $\mathbf{b}_{k,0} = \mathbf{0}$ for $k = 1, \dots, c$, $\mathbf{W}_1 = \mathbf{0}$
for $t = 1$ to T **do**
 Receive $\mathbf{m}_t \in \mathbb{R}^d$
 Predict $\hat{y}_t = \arg \max_k (\mathbf{w}_{k,t}^\top \mathbf{m}_t + \eta_t \sqrt{\mathbf{m}_t^\top \mathbf{A}_{k,t-1}^{-1} \mathbf{m}_t})$
 if $M_t = 1$ **then**
 $\mathbf{b}_{\hat{y}_t,t} = \mathbf{b}_{\hat{y}_t,t-1} - \mathbf{m}_t$
 else
 $\mathbf{b}_{\hat{y}_t,t} = \mathbf{b}_{\hat{y}_t,t-1} + \mathbf{m}_t$
 end if
 $\mathbf{A}_{\hat{y}_t,t} = \mathbf{A}_{\hat{y}_t,t-1} + \mathbf{m}_t \mathbf{m}_t^\top$
 $\mathbf{w}_{\hat{y}_t,t+1} = \mathbf{A}_{\hat{y}_t,t}^{-1} \mathbf{b}_{\hat{y}_t,t}$
end for

It can be seen that, different from online learning in the full information setting, where the learner are able to adjust all the c weight vectors because the true class label y_t is disclosed, the learner in the partial information setting can only update $\mathbf{w}_{\hat{y}_t}$ because y_t is unknown unless $M_t = 0$. This explains why OSLG can utilize more label information. However, since OSLG Bandit only updates one weight vector instead of c , OSLG Bandit turns out to be computationally more efficient.

B. Theoretical Analysis

Our main theoretical result provides bounds on the cumulative regret for Algorithm 2. We adapted the proof technique used in [7]. We will bound the extent to which the prediction error of our learning algorithm exceeds the prediction error of the Bayes optimal predictor

$$y_t^* = \arg \max_{k=1, \dots, c} P(Y_t = k | \mathbf{m}_t) = \arg \max_{k=1, \dots, c} \mathbf{u}_k^\top \mathbf{m}_t,$$

for this label noise model. In particular, we aim to bound from above the cumulative regret

$$R_T = \sum_{t=1}^T (P_t(y_t \neq \hat{y}_t) - P_t(y_t \neq y_t^*)), \quad (3)$$

where P_t denotes the conditional probability as follows

$$P_t(\cdot) = P(\cdot | \mathbf{m}_1, \dots, \mathbf{m}_t, y_1, \dots, y_{t-1}).$$

Note that \mathbf{m}_t can be chosen adversarially as a function of past \mathbf{m} and y .

Theorem 2 (Regret Bound of OSLG Bandit) Let $S = \{(\mathbf{m}_1, y_1), \dots, (\mathbf{m}_T, y_T)\} \in (\mathbb{R}^d \times \{1, \dots, c\})^T$. If we set

$$\eta_t = \mu \|\mathbf{U}\|_2^2 + 4cd \log \left(1 + \frac{(t-1)B^2}{cd\mu} \right) + 72 \log \frac{t+4}{\delta}$$

Then for any $\mathbf{U} \in \mathbb{R}^{d \times c}$ such that $|\mathbf{u}_k^T \mathbf{m}_t| \leq 1$, $\mathbf{F} \in \{-1, 1\}^{T \times c}$, Algorithm 2 has

$$R_T \leq \sqrt{T} \left[\mu \Phi(\mathbf{F}) + \left(\frac{\mu+1}{2\mu} + 2cd \right) \log \left(1 + \frac{TB^2}{cd\mu} \right) + 36 \log \frac{T+4}{\delta} \right]$$

with probability at least $1-\delta$ uniformly over the time horizon T .

Note that the regret bound of OSLG Bandit is $O(cd\sqrt{T} \log T)$, which is only a \sqrt{T} factor worse than that of OSLG, i.e., $O(cd \log T)$. In addition, the difficulty of online learning on a graph with bandit feedback is also characterized by the (minimum) graph cut size.

IV. EXPERIMENTAL RESULTS

In this section, we empirically evaluate the effectiveness of the proposed algorithms.

A. Data Sets

Due to the space limit, we only used two real-world graph data sets to evaluate the online learning algorithms, which have been used in [8].

Coauthor² is an undirected co-author graph data set extracted from the DBLP database in four areas: *machine learning*, *data mining*, *information retrieval* and *databases*.

IMDB³ is an undirected co-actor graph among movies from four genres: “Romance”, “Action”, “Animation” and “Thriller”. Each genre is considered as a class.

The detailed statistics of these data sets can be found in [8].

B. Evaluation Measures

We evaluated the performance of online learning via two measures: (i) cumulative error rate, which reflects the prediction performance of online learning algorithms; and (ii) computational time, which measures the efficiency of online learning. Note that the smaller the above measures, the better the performance of an online learning algorithm.

C. Baselines and Parameter Settings

We compare the proposed algorithms with the Graph Perceptron Algorithm (GPA) [11]. The algorithms we studied and their parameter settings are summarized as follows.

Graph Perceptron Algorithm (GPA) [11]: This is the state-of-the-art first-order online learning algorithm on graphs. There is no required parameter for this algorithm.

²<http://www.imdb.com/>

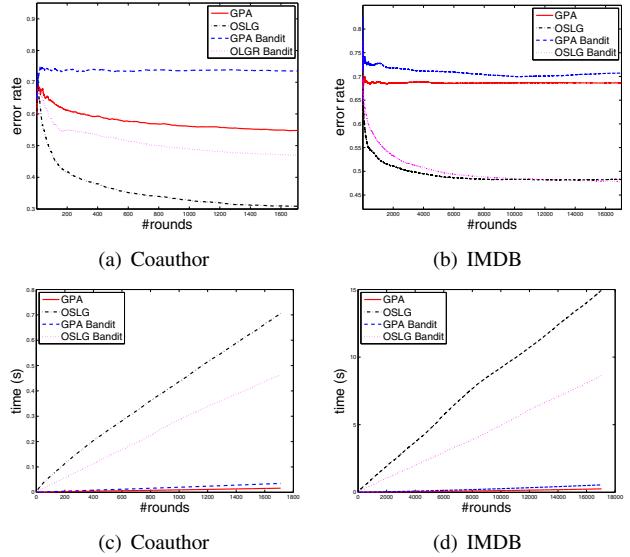


Figure 1. Error rate (top row), and cumulative time (bottom row) with respect to the online learning rounds on Coauthor and IMDB datasets. The lower the curve, the better the performance is.

Note that the original GPA algorithm [11] is designed for binary classification. However, it can be extended to multi-class classification straightforwardly using standard technique, e.g., [15].

Online Spectral Learning on a Graph (OSLG): This is the proposed second-order online learning algorithm on a graph. The parameter μ is tuned by searching the grid $\{10^{-3}, 10^{-2}, \dots, 10\}$ on a held-out random shuffle. The number of eigenvectors used for spectral learning is set to $d = 100$.

GPA Bandit: This is an extension of GPA to the bandit feedback setting by the mechanism designed in [15]. There is a parameter $\gamma \in (0, 0.5)$ which controls the probability of exploration. It is tuned by searching the grid $\{10^{-3}, 10^{-2}, 10^{-1}, 0.5\}$ on a held-out random shuffle.

OSLG Bandit: This is the proposed bandit algorithm on a graph. The parameter μ is tuned according to the grid $\{10^{-3}, 10^{-2}, \dots, 10\}$ on a held-out random shuffle. In our experiments, we fix η_t for all t and tune it by the grid $\{10^{-3}, 10^{-2}, \dots, 10\}$ on a held-out random shuffle as well. The number of eigenvectors used for spectral learning is also set to $d = 100$.

In order to compare these algorithms fairly, we randomly shuffle the ordering of nodes for each dataset. We repeat each experiment 20 times and calculate the average results.

D. Results and Discussions

We show the results with respect to the round of online learning in Figure 1. In all subfigures, the horizontal-axis represents the rounds of online learning, while the vertical-axis is the cumulative error rate and cumulative time, averaged over 20 runs.

We can see that OSLG outperforms GPA significantly on every data set. This is consistent with previous studies: second-order algorithms are generally better than first-order algorithms [8] [13]. Unsurprisingly, OSLG requires more time than GPA. The reason is that the time complexity of GPA is $O(d)$, while the time complexity of OSLG is $O(d^2)$. However, provided that d is moderate, the relatively high computational time of OSLG is acceptable and overall it is more appealing than GPA.

OSLG Bandit is better than GPA and GPA Bandit. This justifies the superiority of our algorithm over GPA Bandit. Moreover, OSLG Bandit achieves slightly worse results than OSLG. This is quite reasonable, because OSLG Bandit only uses partial label information in each round, while OSLG uses the full label information in each round. Thus, the performance of OSLG Bandit should be no better than that of OSLG.

Furthermore, it can be seen that OSLG Bandit takes less time than OSLG. The reason is obvious: OSLG Bandit only updates one weight vector in each round, while OSLG needs to update c weight vectors provided that a mistake is made. Therefore, OSLG Bandit is computationally more efficient than OSLG overall.

In summary, OSLG Bandit is both computational efficient and label information efficient.

V. CONCLUSIONS

The main contributions of this paper are two-fold: (i) we present a multi-class online classification algorithm on a graph using spectral learning technique, and analyze its regret bound; (ii) we present a multi-class online classification algorithm on a graph with bandit feedback, and show that its regret bound is only a \sqrt{T} factor worse than its counterpart in the full information setting.

VI. ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their helpful comments. Research was sponsored in part by the Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), the Army Research Office under Cooperative Agreement No. W911NF-13-1-0193, National Science Foundation IIS-1017362, IIS-1320617, and IIS-1354329, HDTRA1-10-1-0120, and MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC.

REFERENCES

- [1] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.
- [2] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.
- [3] N. Cesa-Bianchi, A. Conconi, and C. Gentile. A second-order perceptron algorithm. *SIAM J. Comput.*, 34(3):640–668, 2005.
- [4] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.
- [5] W. Chu, L. Li, L. Reyzin, and R. E. Schapire. Contextual bandits with linear payoff functions. In *AISTATS*, pages 208–214, 2011.
- [6] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, February 1997.
- [7] K. Crammer and C. Gentile. Multiclass classification with bandit feedback using adaptive regularization. *Machine Learning*, 90(3):347–383, 2013.
- [8] Q. Gu, C. C. Aggarwal, J. Liu, and J. Han. Selective sampling on graphs for classification. In *KDD*, pages 131–139, 2013.
- [9] M. Herbster and G. Lever. Predicting the labelling of a graph via minimum $\$p\$$ -seminorm interpolation. In *COLT*, 2009.
- [10] M. Herbster, G. Lever, and M. Pontil. Online prediction on large diameter graphs. In *NIPS*, pages 649–656, 2008.
- [11] M. Herbster and M. Pontil. Prediction on a graph with a perceptron. In *NIPS*, pages 577–584, 2006.
- [12] M. Herbster, M. Pontil, and L. Wainer. Online learning over graphs. In *ICML*, pages 305–312, 2005.
- [13] S. C. H. Hoi, J. Wang, and P. Zhao. Exact soft confidence-weighted learning. In *ICML*, 2012.
- [14] M. Ji, T. Yang, B. Lin, R. Jin, and J. Han. A simple algorithm for semi-supervised learning with improved generalization error bound. In *ICML*, 2012.
- [15] S. M. Kakade, S. Shalev-Shwartz, and A. Tewari. Efficient bandit algorithms for online multiclass prediction. In *ICML*, pages 440–447, 2008.
- [16] S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.
- [17] A. J. Smola and R. I. Kondor. Kernels and regularization on graphs. In *COLT*, pages 144–158, 2003.
- [18] S. Wang, R. Jin, and H. Valizadegan. A potential-based framework for online multi-class learning with partial feedback. *AISTATS*, pages 900–907, 2010.
- [19] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, 2003.
- [20] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.
- [21] X. Zhu, J. S. Kandola, Z. Ghahramani, and J. D. Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In *NIPS*, 2004.