Min-Max Optimization for Bandit Online Graph Classification

Abstract

Traditional online learning on graph adapts graph Laplacian into ridge regression, which may not guarantee reasonable accuracy when the data is adversarially generated. To solve this issue, we exploit a min-max optimization framework for online classification on graph. The derived algorithm can achieve a min-max regret under an adversarial setting of data generation. To take advantage of the informative labels, we propose an adaptive large-margin update rule, which enjoys a lower regret than the algorithms using error-driven update rules. Although this algorithm is effective, it assumes that the full information label is provided for each node. However, in many practical applications, labeling is expensive and the oracle may only tell whether the prediction is correct for each round of prediction. To address this issue, we propose an online algorithm on graph with bandit feedback. It derives per-instance confidence region of the prediction, from which the model is learned adaptively, and the exploration-exploitation tradeoff is utilized in an informed way. Experiments on benchmark graph datasets show that the proposed bandit algorithm outperforms state-of-the-art competitors, even sometimes beats the algorithms using full information label feedback.

1 Introduction

Existing online classification models on graph usually adapt the graph Lapalacian regularization into ridge regression [Herbster and Pontil, 2006; Gu *et al.*, 2013], that may not guarantee sufficiently high performance in a non-stationary setting where the label for any node (i.e. $\mathbf{x}_t$) is adversarially generated [Vovk, 1998]. To solve this problem, online learning has been extensively studied in the adversarial environment [Moroshko *et al.*, 2015; Kuznetsov and Mohri, 2016]. Although those methods are applicable, they assume that the true label is provided for each node, which violates many real-world scenarios where labeling is very expensive and the label oracle may return partial label feedback for each prediction. To address this issue, online multi-class classification with *bandit feedback* has been studied recently: in each round, after predicting the label of a given node, the learner receives a binary feedback indicating whether the predicted label is correct or not. Although these methods are effective to solve the bandit problem, they either are designed for a nondeterministic exploration-exploitation tradeoff [Crammer and Gentile, 2013; Beygelzimer *et al.*, 2017] or exploit an error-driven update rule instead of an adaptive update strategy [Gu and Han, 2014].

In this paper, based on the observation above, we present a min-max optimization approach for online adaptive multiclass classification on graph. The derived online algorithm can bound the choices of an adversary under a stochastic setting of data generation [Abernethy *et al.*, 2009; Vovk, 1998]. In particular, we propose an adaptive large-margin update rule to prioritize the informative labels. The theoretical result shows that the proposed algorithm, thanks to the aggressive update trials, can achieve a lower error bound than the the algorithms using error-driven update rules. However, this algorithm assumes that the true label is provided for every node, which violates many practical scenarios where labeling is expensive and label oracle may only give an one-bit feedback indicating whether the prediction is correct. To solve this issue, we propose a bandit online algorithm for graph node classification. It constructs per-instance confidence region from partial label feedback, from which the model can be learned adaptively and the exploration-exploitation tradeoff is maintained in an informed way. Experiments on several benchmark graph datasets show that the proposed bandit algorithm outperforms several competitors, even sometimes beats state-of-the-art algorithms using full label feedback.

Notation. We use lower case letter as scalar (e.g. f), bold-face lower letter as vector (e.g. $\mathbf{w}$), upper case letter as element of a matrix (e.g. X_{ij}) and bold-face upper letter as matrix (e.g. $\mathbf{A}$). With an appropriate size, an identity matrix is denoted as $\mathbf{I}$, zero vector as $\mathbf{0}$ and diagonal matrix as diag$(\cdot)$. The inverse of a matrix $\mathbf{A}$ is denoted as $\mathbf{A}^{-1}$, and the pseudo inverse as $\mathbf{A}^{\dagger}$. Finally, the ℓ_2-norm of vector $\mathbf{w}$ is denoted as $\|\mathbf{w}\|_2$, Frobenius norm of matrix $\mathbf{A}$ as $\|\mathbf{A}\|_F$, trace and determinant of a matrix as tr$(\mathbf{A})$ and $|\mathbf{A}|$, respectively.

2 Framework of Graph Node Classification

A graph defined by $G = (V, E)$ consists of an vertex set $V = \{v_i\}_{i=1}^n$, and an edge set $E = \{e_{ij} = \{v_i, v_j\} | v_i, v_j \in V\}$.

An adjacency matrix of the graph G is denoted as $\mathbf{S} \in \mathbb{R}^{n\times n}$, where the element value S_{ij} is computed based on the edge affinity of a pair $\{v_i, v_j\}$. We assume that graph G is connected and undirected. Graph Laplacian is defined by $\mathbf{L} = \mathbf{D} - \mathbf{S}$, where $\mathbf{D}$ is a diagonal matrix with $D_{ii} = \sum_k S_{ik}$ while the off-diagonal elements are 0. The eigendecomposition of graph Laplacian is $\mathbf{L} = \mathbf{V\Lambda V}^\top$, where $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ are the eigenvectors, and $\mathbf{\Lambda}$ = diag$(\lambda_1, \ldots, \lambda_n)$ (0 =$\lambda_1 \leq \ldots \leq \lambda_n \leq 2$) are the eigenvalues.

Graph regularization [Smola and Kondor, 2003] is built based on the concept of label smoothness on graph, i.e., $\sum_{i,j=1}^n S_{ij}\|\mathbf{f}_i - \mathbf{f}_j\|^2 = \text{tr}(\mathbf{F}^\top\mathbf{LF})$, where $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_n]^\top$ ($\mathbf{f}_i \in \mathbb{R}^K$) is a prediction matrix. We solve the problem of multi-class classification on graph as below,

$$\min_{\mathbf{F}} \sum_{i=1}^n \ell(\mathbf{f}_i, \mathbf{y}_i) + b\text{tr}(\mathbf{F}^\top\mathbf{LF}), \tag{1}$$

where $\ell(\mathbf{f}_i, \mathbf{y}_i)$ is the loss suffered by the algorithm on the i-th node, $\mathbf{y}_i \in R^K$ is the actual label of that node, and $b > 0$ is a trade-off parameter. In the multi-class classification with labeled and unlabeled nodes on graph, this optimization is to learn the function $\mathbf{F}$ with two requirements: (i) the predictions for labeled nodes should be close to the given labels for those nodes; (ii) label-smooth on the graph: the nodes nearby on the graph should have similar predictions.

To solve the Eq. (1), we consider its dual form. According to the definition of graph kernel [Smola and Kondor, 2003; Belkin *et al.*, 2006], the function $\mathbf{F}$ is formulated as,

$$\mathbf{F} = \mathbf{L}^\dagger\Omega = \sum_{i=1}^n (\frac{1}{\lambda_i}\mathbf{v}_i\mathbf{v}_i^\top)\Omega, \tag{2}$$

where $\mathbf{L}^\dagger$ is pseudo inverse of $\mathbf{L}$ and $\Omega \in \mathbb{R}^{n\times K}$ is a parameter matrix. We further decompose $\mathbf{L}^\dagger = \sum_i \frac{1}{\lambda_i}\mathbf{v}_i\mathbf{v}_i^\top$ as $\mathbf{L}^\dagger = \mathbf{X}^\top\mathbf{X}$ with $\mathbf{X} = [\frac{1}{\sqrt{\lambda_1}}\mathbf{v}_1, \ldots, \frac{1}{\sqrt{\lambda_n}}\mathbf{v}_n]^\top$, and the kernel Eq. (2) can be rewritten as a linear model form, $\mathbf{F} = \mathbf{X}^\top\mathbf{W}$ where $\mathbf{W} = \mathbf{X}\Omega \in \mathbb{R}^{n\times K}$. Substituting the graph kernel Eq. (2) into Eq. (1) with $\mathbf{XLX}^\top = \mathbf{I}$, we have

$$\min_{\mathbf{W}} \sum_{i=1}^n \ell(\mathbf{W}; (\mathbf{x}_i, \mathbf{y}_i)) + b\|\mathbf{W}\|_F^2. \tag{3}$$

In this way, we derive an objective function similar to ridge regression formulation. Eq. (3) can be used to derive the online learning under a new graph node representation.

3 Online Learning on Graph

The objective of online learning on graph is to achieve a low regret compared with the best linear model in hindsight. Let $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)\}$ ($T \leq n$) be an arbitrary node-label sequence. At round t, online algorithm receives a node $\mathbf{x}_t$, and predicts its label: $\hat{y}_t = \arg\max_{i\in[K]}(\mathbf{W}_{t-1}^\top\mathbf{x}_t)_i$, where the model $\mathbf{W}_{t-1}$ is learned from the previous $t-1$ rounds. Then the actual label y_t is revealed, the algorithm uses it to update model and then proceeds to the next round. For any matrix $\mathbf{U} \in \mathbb{R}^{n\times K}$, we denote by $\ell_t(\mathbf{U}) = \|\mathbf{U}^\top\mathbf{x}_t - \mathbf{y}_t\|^2$ as the instantaneous loss of $\mathbf{x}_t$, and by $L_T(\mathbf{U}) = \sum_{t=1}^T \ell_t(\mathbf{U})$ as the cumulative loss over T rounds. Formally, we define the regret of the algorithm to be

$$R_T(\mathbf{U}) = \sum_{t=1}^T \|\mathbf{W}_{t-1}^\top\mathbf{x}_t - \mathbf{y}_t\|^2 - \inf_{\mathbf{U}} L_T(\mathbf{U}),$$

The objective is to let the cumulative loss of the online algorithm converge to the loss of the best linear function in hindsight. We next solve the regret in an adversarial setting.

3.1 Min-max Online Prediction on Graph

In the adversarial scenario, the regret is solved based on a last-step min-max optimization [Forster, 1999; Takimoto and Warmuth, 2000]. An algorithm following this objective outputs a min-max prediction assuming that the current iteration is the last one, while $\hat{y}_t$ and y_t serve as min and max quantifiers, i.e., the goal of the learner is to minimize the regret while the adversary is to maximize it. Note that our objective function is different from [Forster, 1999; Takimoto and Warmuth, 2000], since the model is derived from graph regularization. For this purpose, we introduce an adaptive-weighted cumulative loss,

$$L_T^{\mathbf{a}}(\mathbf{U}) = \sum_{t=1}^T a_t\|\mathbf{y}_t - \mathbf{U}^\top\mathbf{x}_t\|^2, \tag{4}$$

where $\{a_t\}_{t=1}^T \geq 0$ are the input-dependent weights. Substituting Eq. (3) and Eq. (4) into the regret $R_T(\mathbf{U})$, our variant of the min-max algorithm predicts,

$$\min_{\hat{y}_T}\max_{y_T} \sum_{t=1}^T \|\mathbf{f}_t - \mathbf{y}_t\|^2 - \inf_{\mathbf{U}}(b\|\mathbf{U}\|_F^2 + L_T^{\mathbf{a}}(\mathbf{U})), \tag{5}$$

with $\mathbf{f}_t = \mathbf{W}_{t-1}^\top\mathbf{x}_t$. We next solve the min-max adaptive-weighted function Eq. (5), starting with additional notations,

$$\mathbf{A}_t = b\mathbf{I} + \sum_{s=1}^t a_s\mathbf{x}_s\mathbf{x}_s^\top,\ \mathbf{B}_t = \sum_{s=1}^t a_s\mathbf{x}_s\mathbf{y}_s^\top, \tag{6}$$

The solution of the internal infimum is obtained as below.

Lemma 1 *For all* $t \geq 1$, $G_t(\mathbf{U}) = L_t^{\mathbf{a}}(\mathbf{U}) + b\|\mathbf{U}\|_F^2$ *is minimal at a unique point* $\mathbf{U}$*, given by,*

$$\mathbf{U} = \mathbf{A}_t^{-1}\mathbf{B}_t,\ \inf_{\mathbf{U}} G_t(\mathbf{U}) = \sum_{s=1}^t a_s\|\mathbf{y}_s\|^2 - tr(\mathbf{B}_t^\top\mathbf{A}_t^{-1}\mathbf{B}_t).$$

Substituting Lemma 1 back into Eq. (5), we reformulate the objective function, $\hat{y}_T = \min_{\hat{y}_T}\max_{y_T} F(\mathbf{y}_t, \mathbf{f}_t)$, where

$$F(\mathbf{y}_t, \mathbf{f}_t) = \sum_{t=1}^T (\|\mathbf{f}_t - \mathbf{y}_t\|^2 - a_t\|\mathbf{y}_t\|^2) + \text{tr}(\mathbf{B}_T^\top\mathbf{A}_T^{-1}\mathbf{B}_T).$$

We show that, by appropriately setting the weight a_T, the min-max problem could be concave in y_T and convex in $\hat{y}_T$. The optimal solution of the min-max predictor is summarized as follows.

Theorem 1 *Assume that* $1 + a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T - a_T \leq 0$, *then the optimal prediction of the min-max function* $\hat{y}_T = \arg\min_{\hat{y}_T}\max_{y_T} F(\mathbf{y}_T, \mathbf{f}_T)$ *is*

$$\hat{y}_T = \arg\max_{i\in[K]}(\mathbf{B}_{T-1}^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T)_i. \tag{7}$$

Proof. Since $\mathrm{tr}(\mathbf{B}_T^\top \mathbf{A}_T^{-1} \mathbf{B}_T) = \mathrm{tr}(\mathbf{B}_{T-1}^\top \mathbf{A}_T^{-1} \mathbf{B}_{T-1}) + 2a_T \mathbf{B}_{T-1}^\top \mathbf{A}_T^{-1} \mathbf{x}_T \cdot \mathbf{y}_T + a_T^2 \mathbf{x}_T^\top \mathbf{A}_T \mathbf{x}_T \|\mathbf{y}_T\|^2$, we obtain by omitting unrelated terms, $\min_{\hat{y}_T} \max_{y_T} \; F(\mathbf{y}_T, \mathbf{f}_T) = \alpha(a_T)\|\mathbf{y}_T\|^2 + 2\beta(a_T, \mathbf{f}_T) \cdot \mathbf{y}_T + \|\mathbf{f}_T\|^2$, where $\alpha(a_T) = \frac{1+a_T \mathbf{x}_T^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T - a_T}{1+a_T \mathbf{x}_T^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T}$, $\beta(a_T, \mathbf{f}_T) = a_T \mathbf{B}_{T-1}^\top \mathbf{A}_T^{-1} \mathbf{x}_T - \mathbf{f}_T$. We consider two cases: (1) $1 + a_T \mathbf{x}_T^\top \mathbf{A}_{T-1}^\top \mathbf{x}_T - a_T < 0$ and (2) $1 + a_T \mathbf{x}_T^\top \mathbf{A}_{T-1}^\top \mathbf{x}_T - a_T = 0$. Starting with the first case, the function $F(\mathbf{y}_T, \mathbf{f}_T)$ is strictly-concave with respect to $\mathbf{y}_T$ since $\alpha(a_T) < 0$. Thus, $F(\mathbf{y}_T, \mathbf{f}_T)$ attains a unique maximal value when $\mathbf{y}_T = -\frac{\beta(a_T,\mathbf{f}_T)}{\alpha(a_T)}$, i.e., $\min_{\hat{y}_T} F(-\frac{\beta(a_T,\mathbf{f}_T)}{\alpha(a_T)}, \mathbf{f}_T) = \frac{-a_T}{1+a_T \mathbf{x}_T^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T - a_T} \|\mathbf{f}_T\|^2 + \frac{2a_T \mathbf{B}_{T-1}^\top \mathbf{A}_T^{-1} \mathbf{x}_T (1+a_T \mathbf{x}_T^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T)}{1+a_T \mathbf{x}_T^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T - a_T} \cdot \mathbf{f}_T - \gamma(a_T)$. This objective is strictly-convex with respect to $\mathbf{f}_T$, and easy to obtain the optimal predictor, $\mathbf{f}_T = \mathbf{B}_{T-1}^\top \mathbf{A}_T^{-1} \mathbf{x}_T (1 + a_T \mathbf{x}_T^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T) = \mathbf{B}_{T-1}^\top \mathbf{A}_T^{-1} \mathbf{x}_T + \mathbf{B}_{T-1}^\top \mathbf{A}_T^{-1} (\mathbf{A}_T - \mathbf{A}_{T-1}) \mathbf{A}_{T-1}^{-1} \mathbf{x}_T = \mathbf{B}_{T-1}^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T$. Considering the second case for which, $1 + a_T \mathbf{x}_T^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T - a_T = 0$, the problem can be formulated as $\min_{\hat{y}_T} \max_{y_T} 2(a_T \mathbf{B}_{T-1}^\top \mathbf{A}_T^{-1} \mathbf{x}_T - \mathbf{f}_T) \cdot \mathbf{y}_T + \|\mathbf{f}_T\|^2$. If $\mathbf{f}_T \neq a_T \mathbf{B}_{T-1}^\top \mathbf{A}_T^{-1} \mathbf{x}_T$, the objective value is not-bounded. Thus, the optimal min-max prediction is $\mathbf{f}_T = a_T \mathbf{B}_{T-1}^\top \mathbf{A}_T^{-1} \mathbf{x}_T = \mathbf{B}_{T-1}^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T$, given $a_T = \frac{1}{1-\mathbf{x}_T^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T}$, and $\mathbf{A}_t^{-1} = \mathbf{A}_{t-1}^{-1} - \mathbf{A}_{t-1}^{-1} \mathbf{x}_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}$. □

Remark. Although the solution is applicable for $1 + a_T \mathbf{x}_T^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T - a_T \leq 0$, we make $a_T = \frac{1}{1-\mathbf{x}_T^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T}$ in the rest of paper. The a_T dependents on previous learned rounds $\{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ with a fixed computational cost $O(n^2)$.

3.2 Algorithm and Theoretical Analysis

We summarize the algorithm, the Min-max Optimization for Online Learning on Graph (defined by (6) and (7)), in Algorithm 1. It maintains two quantities: $\mathbf{A}_t$ and $\mathbf{B}_t$. Inspired by [Azoury and Warmuth, 2001], we exploit current input to predict its label $\mathbf{f}_t = \mathbf{B}_{t-1}^\top \mathbf{A}_t^{-1} \mathbf{x}_t$ where $\mathbf{A}_t = \mathbf{A}_{t-1} + a_t \mathbf{x}_t \mathbf{x}_t^\top$. Then its actual label y_t is revealed and the algorithm updates model in terms of $\mathbf{B}_t$ and $\mathbf{A}_t$ in a recursive way: $\mathbf{B}_t = \mathbf{B}_{t-1} + a_t \mathbf{x}_t \mathbf{y}_t^\top$, and $\mathbf{A}_t^{-1} = \mathbf{A}_{t-1}^{-1} - \mathbf{A}_{t-1}^{-1} \mathbf{x}_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}$ according to Woodbury identity. In each round of update, the runtime is $O(n^2)$ and the required memory is $O(n^2)$.

Existing online algorithms update model when an error occurs (e.g., $\triangle_t = \mathbf{f}_t \cdot \mathbf{y}_t \leq 0$), which ignore exploiting the correctly predicted labels of low prediction confidence to improve the model accuracy. To solve this issue, we propose an adaptive large-margin update rule. We define the prediction margin on round t in multi-class classification,

$$\hat{\Delta}_t = \mathbf{f}_t \cdot \mathbf{y}_t = (\mathbf{W}_{t-1}^\top \mathbf{x}_t)_{y_t} - \max_{i \neq y_t} (\mathbf{W}_{t-1}^\top \mathbf{x}_t)_i, \quad (8)$$

where the label vector $\mathbf{y}_t \in \mathbb{R}^K$ assigns $+1$ to the true class entry y_t, -1 to the class entry $j = \arg\max_{i \neq y_t} (\mathbf{W}_{t-1}^\top \mathbf{x}_t)_i$, and 0 to the remain entries.

Definition 1 *Given an input $\mathbf{x}_t$ ($t \in [T]$), the algorithm predicts its label $\mathbf{f}_t = \mathbf{W}_{t-1}^\top \mathbf{x}_t$ and updates model whenever the function $\Theta_t < 0$, where Θ_t is the proposed adaptive margin towards current prediction,*

$$\Theta_t = \hat{\Delta}_t - \sigma_t, \quad (9)$$

where $\sigma_t = \frac{1}{2} a_T^2 \mathbf{x}_T^\top \mathbf{A}_T^{-1} \mathbf{x}_T$, and $\hat{\Delta}_t$ is denoted in Eq. (8).

Algorithm 1 MOLG-F: Adaptive Min-max Optimization for Online Learning on Graph with Full Label Feedback

```
Input: Adjacency matrix $\mathbf{S}$, rank $d$, and regularization
  parameters $b$ and $\phi$.
Output: $\mathbf{W}_T$
Compute $\mathbf{L} = \mathbf{D} - \mathbf{S}$ and $\mathbf{X}$ from $\mathbf{L}^\dagger$
Initialize: $\mathbf{W}_0 = \mathbf{0}, \mathbf{B}_0 = \mathbf{0}, \mathbf{A}_0 = b\mathbf{I}$;
for $t = 1, \dots, T$ do
    Receive $(\mathbf{x}_t, y_t)$
    $\mathbf{A}_t^{-1} = (\mathbf{A}_{t-1} + a_t \mathbf{x}_t \mathbf{x}_t^\top)^{-1}, \mathbf{W}_{t-1} = \mathbf{A}_t^{-1} \mathbf{B}_{t-1}$;
    Predict $\hat{y}_t = \arg\max_{i \in [K]} (\mathbf{W}_{t-1}^\top \mathbf{x}_t)_i$
    if $\hat{y}_t \neq y_t$ then $Z_t = 1$
    else
        $\Theta_t = \hat{\Delta}_t - \phi \sigma_t$;
        if $\Theta_t < 0$ then $Z_t = 1$
        end if
    end if
    Update $\mathbf{A}_t^{-1} = (\mathbf{A}_{t-1} + Z_t a_t \mathbf{x}_t \mathbf{x}_t)^{-1}$;
    Update $\mathbf{B}_t = \mathbf{B}_{t-1} + Z_t a_t \mathbf{x}_t \mathbf{y}_t^\top$;
end for
```

The Θ_t is a function parameterized by $\hat{\Delta}_t$ and σ_t, where $\hat{\Delta}_t$ is the *prediction margin*, and σ_t is the *confidence region* of the prediction. In this way, $\Theta_t = \hat{\Delta}_t - \sigma_t$ acts as the *lower confidence bound* of current prediction. Note that the adaptive margin technique has been studied in [Wang *et al.*, 2012; Yang *et al.*, 2016]. Different from previous techniques, the proposed Θ_t is derived from the min-max optimal predictor for graph classification. We next theoretically analyze the effectiveness of our algorithm.

In Algorithm 1, the update trials can be partitioned into two disjoint sets, $\mathcal{M} = \{t : \hat{\Delta}_t \leq 0, \hat{y}_t \neq y_t\}$ with $M = |\mathcal{M}|$ includes the indices on which a update is issued when an error occurs, and $\mathcal{D} = \{t : 0 < \hat{\Delta}_t < \sigma_t, \hat{y}_t = y_t\}$ with $D = |\mathcal{D}|$ includes the indices on which an aggressive update is issued, even if the prediction is correct. Let $\mathcal{Z} = \{t : Z_t = 1\}$ with $Z = |\mathcal{Z}|$ be a set of update trials containing $Z = M + D$.

Theorem 2 *Algorithm 1 runs on an arbitrary node sequence $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)$. Let $\tilde{\mathcal{L}}(x) = \max(0, 1 - x)$ be hinge loss, for any $\mathbf{U} \in \mathbb{R}^{n \times K}$, the following inequality holds,*

$$M \leq \sum_{t \in \mathcal{Z}} a_t \tilde{\mathcal{L}}(\mathbf{y}_t \cdot \mathbf{U}^\top \mathbf{x}_t) + \frac{1}{2} tr(\mathbf{U}^\top \mathbf{A}_{\mathcal{Z}} \mathbf{U}) + \frac{b}{b-1} \log |\frac{1}{b} \mathbf{A}_T| - D.$$

Proof. The proof of Theorem 2 is in the Appendix. □

Remark. Assume $\|\mathbf{x}\|_2 \leq 1$, given $\mathbf{A}_t^{-1} \preceq \mathbf{A}_{t-1}^{-1}$ and $\mathbf{x}_t^\top \mathbf{A}_t^{-1} \mathbf{x}_t \leq \dots \leq \mathbf{x}_t^\top \mathbf{A}_0^{-1} \mathbf{x}_t \leq \frac{1}{b}$, we have $a_t = \frac{1}{1-\mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1} \mathbf{x}_t} \leq \frac{b}{b-1}$, and $\mathrm{tr}(\mathbf{U}^\top \mathbf{A}_{\mathcal{Z}} \mathbf{U}) \leq$

$\sum_{t\in\mathcal{Z}} a_t\|\mathbf{U}^\top\mathbf{x}_t\|^2 \le \frac{Zb}{b-1}\|\mathbf{U}\|_F^2$. Moreover, $\log|\frac{1}{b}\mathbf{A}_T| \le Kn\log(1+\frac{T}{Knb})$ [Moroshko and Crammer, 2014]. Due to the deduction of the aggressive update trials $|\mathcal{D}|$, the error bound of Algorithm 1 can be lower than that of the OSLG using error-driven update rules [Gu and Han, 2014].

Discussion: $\sum_{t\in\mathcal{Z}} \tilde{\mathcal{L}}(\mathbf{y}_t\cdot\mathbf{U}^\top\mathbf{x}_t)$ is the cumulative hinge loss made by $\mathbf{U}$. We rewrite the error bound as one by which the number of mistakes exceeds the cumulative loss of the best linear model on the update trials,

$$\begin{aligned} &M - \inf_{\mathbf{U}} \sum_{t\in\mathcal{Z}} a_t\tilde{\mathcal{L}}(\mathbf{y}_t\cdot\mathbf{U}^\top\mathbf{x}_t) \\ \le &\frac{Zb}{2(b-1)}\|\mathbf{U}\|_F^2 + \frac{Knb}{b-1}\log(1+\frac{T}{Knb}) - D. \end{aligned}$$

From above inequality, we summarize that our bound can achieve a regret of $O(\log T)$ with respect to the best model.

3.3 Online Learning with Bandit Feedback

We introduce the problem of online classification on graph with bandit feedback: on round t, the learner receives a node $\mathbf{x}_t$ and predicts its label as $\hat{y}_t \in [K]$. Different from Algorithm 1, the learner receives an one-bit feedback $C_t \in \{+1,-1\}$ indicating whether the output $\hat{y}_t$ is correct or not, i.e., $C_t = \mathbb{I}(y_t \neq \hat{y}_t)$. Following the setting of [Crammer and Gentile, 2013], we assume the label of a node $\mathbf{x}_t$ is sampled with a probabilistic model, $P(y_t = i|\mathbf{x}_t) = \frac{1+\Delta_t^i}{2}, i \in [K]$, where $\Delta_t^i = \mathbf{u}^{i\top}\mathbf{x}_t \in [-1,1]$ and $\sum_{i=1}^K \Delta_t^i = 2-K$.

We propose a bandit online algorithm on graph in Algorithm 2. Specifically, the algorithm maintains a set of weight parameters for multiple classes $\{(\mathbf{b}^i \in \mathbb{R}^n, \mathbf{A}^i \in \mathbb{R}^{n\times n})\}_{i=1}^K$, initialized with $\mathbf{b}_0 = \mathbf{0}, \mathbf{A}_0 = b\mathbf{I}$. Provided the one-bit feedback, the basic idea is to maintain a tradeoff between exploration and exploitation [Auer, 2002; Crammer and Gentile, 2013]. Intuitively, the algorithm should output the label with the largest score, i.e., $\arg\max_{i\in[K]} \hat{\Delta}_t^i = \mathbf{w}_{t-1}^{i\top}\mathbf{x}_t$, which is called exploitation. However, if the feedback is negative, the true class is still uncertain. Thus, the algorithm tries to explore the classes with high predicted uncertainties, measured by $\sigma_t^i = \frac{1}{2}(a_t^i)^2\mathbf{x}_t^\top(\mathbf{A}_t^i)^{-1}\mathbf{x}_t$. Generally, a large score of σ_t^i infers a low confidence of current prediction $\hat{\Delta}_t^i$. The bandit algorithm chooses an estimate from the prediction margin and its confidence region, so that the upper bound of the prediction confidence is maximized,

$$\hat{y}_t = \arg\max_{i\in[K]}(\hat{\Delta}_t^i + \varphi_t^i\sqrt{\sigma_t^i}),$$

where $\varphi_t^i > 0$ controls the exploration-exploitation tradeoff. This called upper confidence bound technique has been widely used in online bandit problem [Crammer and Gentile, 2013; Gu and Han, 2014; Beygelzimer *et al.*, 2017]. Our algorithm is different from previous techniques, since MOLG-B is a deterministic algorithm while the Banditron [Crammer and Gentile, 2013; Beygelzimer *et al.*, 2017] is a randomized approach. Moreover, our algorithm uses current input to estimate the prediction margin and prediction confidence, which should be more accurate than OSLG-Bandit [Gu and Han, 2014].

Algorithm 2 MOLG-B: Adaptive Min-Max Optimization for Online Learning on Graph with Bandit Feedback

```
Input: Adjacency matrix $\mathbf{S}$, rank $d$, and regularization parameters $b$, $\varphi$ and $\phi$.
Output: $\mathbf{W}_T$
Compute $\mathbf{L} = \mathbf{D} - \mathbf{S}$ and $\mathbf{X}$ from $\mathbf{L}^\dagger$;
Initialize: $\mathbf{b}_0^i = \mathbf{0}$, $\mathbf{A}_0^i = b\mathbf{I}$ for all $i \in [K]$.
for $t = 1, \ldots, T$ do
    Receive $\mathbf{x}_t$;
    For $i \in [K]$, $(\mathbf{A}_t^i)^{-1} = (\mathbf{A}_{t-1}^i + a_t^i\mathbf{x}_t\mathbf{x}_t^\top)^{-1}$;
    For $i \in [K]$, $\mathbf{w}_{t-1}^i = (\mathbf{A}_t^i)^{-1}\mathbf{b}_{t-1}^i$;
    $\hat{y}_t = \arg\max_{i\in[K]}(\mathbf{w}_{t-1}^{i\top}\mathbf{x}_t + \varphi_t^i\sqrt{\sigma_t^i})$, observe $C_t$;
    if $y_t \neq \hat{y}_t$ then $Z_t = 1$
    else // we denote $k = \hat{y}_t$
        if $\mathbf{w}_{t-1}^k \cdot \mathbf{x}_t < \phi\sigma_t^k$ then $Z_t = 1$;
        end if
    end if
    Update $(\mathbf{A}_t^k)^{-1} = (\mathbf{A}_{t-1}^k + Z_t a_t^k\mathbf{x}_t\mathbf{x}_t^\top)^{-1}$;
    Update $\mathbf{b}_t^k = \mathbf{b}_{t-1}^k + Z_t a_t^k y_t\mathbf{x}_t$;
end for
```

We theoretically analyze the regret of Algorithm 2. It exploits the proofs of the lemmas in [Crammer and Gentile, 2013; Dekel *et al.*, 2010]. Given the Bayes optimal predictor $y_t^* = \arg\max_{i\in[K]}\mathbb{P}(y_t = i|\mathbf{x}_t)$, we aim to bound the regret, i.e., $R_T = \sum_{t=1}^T(\mathbb{P}_t(y_t \neq \hat{y}_t) - \mathbb{P}_t(y_t \neq y_t^*)) = \sum_{t=1}^T \frac{\Delta_t - \hat{\Delta}_t}{2}$, where $\mathbb{P}_t$ is the conditional probability $\mathbb{P}(\cdot|\mathbf{x}_1,\ldots,\mathbf{x}_t,y_1,\ldots,y_t)$.

Theorem 3 *The Algorithm 2 runs on an arbitrary node-label sequence* $\{(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_T,y_T)\}$. *If we set* $\varphi_t^{i2} \le 4(b-1)\|\mathbf{U}\|_F^2 + 4\gamma\log|\frac{1}{b}\mathbf{A}_{t-1}^i| + 144\log\frac{t+4}{\delta}$ *where* $\ell_t(\mathbf{U}) \le \gamma$, *then for any* $\mathbf{U}$, *such that* $|\mathbf{u}^{i\top}\mathbf{x}_t| \le 1$, *the inequality holds,*

$$R_T \le \sqrt{\frac{b}{b-1}T}(\sqrt{H_1H_2} + H_2),$$

with probability at least $1-\delta$ *over* T *trials, where* $H_1 = 2(b-1)\|\mathbf{U}\|_F^2 + 72\log\frac{t+4}{\delta}$ *and* $H_2 = 2Kn\gamma\log(1+\frac{T}{Knb})$.

Proof. The proof will be provided on the website. □

Remark. Theorem 3 shows that the regret bound of MOLG-B is $O(\sqrt{T}\log T)$, which is only a $\sqrt{T}$ factor worse than that of MOLG-F, i.e., $O(\log T)$.

3.4 Low Rank Approximation

Recall that $\mathbf{X} = [\frac{1}{\sqrt{\lambda_1}}\mathbf{v}_1,\ldots,\frac{1}{\sqrt{\lambda_n}}\mathbf{v}_n]^\top$, the graph kernel $\mathbf{F}$ is built exactly, but the time complexity of our algorithm becomes $O(n^2)$, which is computationally expensive on large-sized graphs. In order to make our online algorithm scalable on large graphs, we propose to choose $\mathbf{X}$ as follows,

$$\tilde{\mathbf{X}} = [\frac{1}{\sqrt{\lambda_1}}\mathbf{v}_1,\ldots,\frac{1}{\sqrt{\lambda_d}}\mathbf{v}_d]^\top, \;\; \tilde{\mathbf{W}} = \tilde{\mathbf{X}}\Omega,$$

where $d \ll n$. Thus $\tilde{\mathbf{F}} = (\sum_{i=1}^d \frac{1}{\lambda_i}\mathbf{v}_i\mathbf{v}_i^\top)\Omega$ is a rank-d approximation of $\mathbf{F}$, with a reduced time complexity $O(d^2) \ll O(n^2)$. We analyze the impact of the low-rank approximation on the kernel $\tilde{\mathbf{F}}$. Since $\lambda_1 \leq \ldots \leq \lambda_n$, $\tilde{\mathbf{L}}^\dagger = \sum_{i=1}^{d} \frac{1}{\lambda_i} \mathbf{v}_i \mathbf{v}_i^\top$ retains the top-d largest egivenvalues of $\mathbf{L}^\dagger$. According to Young-Mirsky theorem [Eckart and Young, 1936], we claim that $\tilde{\mathbf{L}}^\dagger$ is the best rank-d approximation of $\mathbf{L}^\dagger$. Thus, $\tilde{\mathbf{F}}$ is the best rank-d approximation of the $\mathbf{F}$.

Equipped with $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n] \in \mathbb{R}^{d \times n}$ where $\tilde{\mathbf{x}}_i = [\frac{1}{\sqrt{\lambda_1}}(\mathbf{v}_1)_i, \ldots, \frac{1}{\sqrt{\lambda_d}}(\mathbf{v}_d)_i]^\top$, the regularized Laplacian term can turn to a sparse form, i.e., $\mathrm{tr}(\tilde{\mathbf{F}}^\top \mathbf{L} \tilde{\mathbf{F}}) = \|\tilde{\mathbf{W}}\|_F^2$, while the graph regularized problem can be formulated as a low-rank form, i.e., $\inf_{\tilde{\mathbf{W}}} \sum_{i=1}^{n} \ell(\tilde{\mathbf{W}}; (\tilde{\mathbf{x}}_i, \mathbf{y}_i)) + b\|\tilde{\mathbf{W}}\|_F^2$.

4 Experimental Results

We begin with data sets and experimental evaluation metrics, and then show empirical results of the proposed algorithms.

4.1 Data Sets and Evaluation Metrics

Data Sets: We exploit 4 real-world graph data sets to evaluate all the algorithms: 1) Coauthor[2] is a coauthor graph of *DBLP* dataset, with a total 1711 authors as nodes and 7507 coauthored associations as edges. These authors are categorized into 4 research topics: "machine learning", "data ming", "database" and "information retrieval". 2) Cora[1] is a citation graph of 2485 scientific publications and 10138 citation links. Each publication is categorized into 7 domains, "Reinforcement Learning", "Neural Networks", "Case based", "Genetic Algorithms", "Probabilistic Methods", "Rule Learning" and "Theory". 3) IMDB[2] is a up-to-date movie dataset with total 17046 motives as nodes and 993528 co-actor relations as edges. Each movie is labeled by one of four genres: "Animation", "Action", "Thriller" and "Romance". 4) PubMed[3] is a graph of 19717 scientific publications of diabetes labeled by one of three types. This graph has 88651 citation links.

All graphs are supposed to be undirected and connected. In case some edges are directed, they are transformed into undirected graphs. If some graphs are disconnected, the biggest connected subgraph is selected for evaluation.

Evaluation Metrics and Parameter Setting: We evaluate the performance of the algorithms with two measures: (i) cumulative error rate, (ii) cumulative update time. Error rate reflects the prediction accuracy of online learning while update time measures the computational efficiency of online algorithm. Note that the smaller the two measures, the better the performance of an online algorithm. In order to compare the algorithms fairly, we randomly shuffle the sample ordering of each dataset. We run all the algorithms 20 times for each dataset and compute the averaged result.

We compare the proposed algorithms with five baselines mentioned in the introduction. The algorithms we studied and their parameter settings are summarized as follows: (a) *GPA* [Herbster and Pontil, 2006] is the first-order nonparametric online learning algorithm on graph. (b) *OSLG* [Gu and Han, 2014] and *MSG* [Yang *et al.*, 2016] are second-order online algorithms on graph with full information label feedback. (c) two variants of *Banditron*, i.e., Confidit [Crammer and Gentile, 2013] and SOBA [Beygelzimer *et al.*, 2017], and *OSLG-Bandit* [Gu and Han, 2014] are second-order online bandit algorithms. All parameters of the baselines are tuned according to their default settings. (d) *MOLG-F* and *MOLG-B* are the proposed online algorithms for graph node classification with full information and bandit label feedback, respectively. For both methods, we tune the parameter ϕ with the grid $\{10^{-2}, \ldots, 10\}$. For MOLG-B, we fix the exploration parameter $\varphi_t = 0.05$ for all $t \in [T]$. We set $b = 10$ for Cora and Coauthor and $b = 100$ for IMDB and PubMed due to variable graph structure. Finally, we fix $d = 100$ for the dimension of low-rank representation.

4.2 Empirical Results

The experimental results are presented in Table 1. The improvement over GPA is always significant on four datasets. This is consistent with previous observations in online learning: the second-order algorithms are generally better than first-order algorithms [Wang *et al.*, 2012]. The reason is that the covariance matrix $\mathbf{A}_t$ that encodes the confidence of parameters can guide the direction of the parameter update in the learning process.

MOLG-B always enjoys fewer update time and smaller error rates than OSLG-Bandit and Banditron (P-value < 0.01). The reasons are two fold: (1) adaptive margin is used to aggressively prioritize the informative label; (2) adaptive confidence is used to guide the magnitude of parameter update. In this way, update time and error rate can be reduced further when the model has learned sufficient knowledge of data. It demonstrates both the computational efficiency and label information efficiency of our bandit algorithm.

MOLG-B achieves slightly worse results than MOLG-F. This is reasonable, since MOLG-B uses partial information label, while MOLG-F uses the full information label. Thus, the performance of the bandit algorithm should be no better than that of the algorithm in full information setting. However, MOLG-F requires more updates than MOLG-B to achieve better accuracy. The reason is obvious: provided one-bit feedback, MOLG-B can choose one class to update its parameter in each round, while MOLG-F can exploit the full information to update multiple-class weights. Overall, MOLG-B is computational more efficient than MOLG-F.

4.3 Sensitivity Study on Update Ratio

In the proposed update rule, the ϕ is a tradeoff parameter of adaptive margin. We tune the parameter ϕ to study the impact of update ratio on the model accuracy. The algorithm with a small ϕ would perform few number of updates. Specifically, we set ϕ to $\{10^{-2}, \ldots, 10\}$ and calculate the averaged ratio of update over the 20 times of experiments. The comparison result is shown in Fig.1.

We observe that MOLG-F achieves the best performance consistently under different ratios of update. Note that B-BQ [Agarwal, 2013] uses an error-driven update rule, thus ignoring all the correctly predicted labels. The empirical results show that updating these labels with low prediction confident is effective to improve the performance.

[1] http://www.cs.umd.edu/ sen/lbc-proj/data/

[2] http://www.imdb.com/

[3] http://www.cs.umd.edu/projects/linqs/projects/lbc/

Table 1: Performance in terms of mean and standard deviation of two measures (Banditron represents the best result from Confidit and SOBA)

Algorithm	***Coauthor***			***Cora***	
	Error rate (%)	# Updated times		Error rate (%)	# Updated times
GPA	54.74±2.66	1711		58.49±2.03	2485
MSG	29.53±0.97	1774±42.48		18.95±0.44	3810.8±37.49
OSLG	31.02±0.55	2123±37.66		20.77±0.31	3612.7±54.13
MOLG-F	**27.86±0.50**	1739±22.77		**18.16±0.39**	2629.7±25.99
Banditron	39.18±3.34	1711		41.49±0.53	2485
OSLG-Bandit	33.81±1.85	1711		28.56±1.54	2485
MOLG-B	32.45±0.92	**1122.5±16.90**		23.87±1.48	**1363.8±27.64**

Algorithm	***IMDB***			***PubMed***	
	Error rate (%)	# Updated times		Error rate (%)	# Updated times
GPA	68.70±0.46	17046		57.95±0.29	19717
MSG	50.89±0.29	29788±173.8		22.64±0.10	27336±128.9
OSLG	48.27±0.60	32910±409.6		25.51±0.39	15087±231.4
MOLG-F	45.86±0.91	22147±336.3		20.96±0.07	17198±28.63
Banditron	52.88±0.38	17046		24.53±0.15	19717
OSLG-Bandit	46.91±0.39	17046		21.07±0.18	19717
MOLG-B	**45.30±1.17**	**13316.7±259.2**		**20.74±0.18**	**13633±112.2**

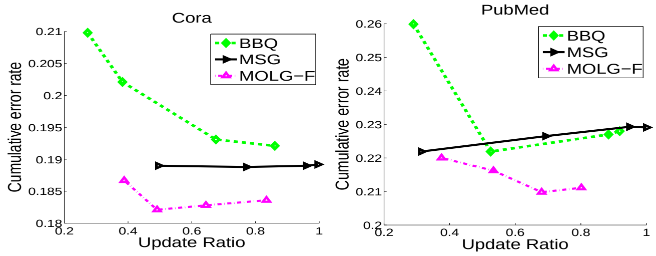

Figure 1: Sensitivity study of updating ratio.

4.4 Sensitivity Study on Low-rank Approximation

The empirical result is affected by the rank approximation. To study the impact of the low-rank representation, we set the parameter d with the grid $\{10, 100, 250, 500, 750, 1000\}$. Coauthor and IMDB are used as case study since similar conclusions are obtained on Cora and PubMed. The results in Fig. 2 show that MOLG-F and MOLG-B outperform OSLG and OSLG-Bandit, respectively, under various low-rank approximation. With a high rank, all the algorithms achieve a lower error rate. However, the algorithm needs a high computational time to update model with time complexity $O(d^2)$. To achieve a balance, we choose $d = 100$ in this paper since the algorithm achieves a good accuracy while number of update is small enough.

5 Conclusion

We explore a min-max optimization for adaptive online graph classification with full information and bandit label feedback. The main contributions are three-fold: (i) the algorithm derived from the min-max optimization can guarantee the performance when the data is adversarially generated; (ii) we derive per-instance confidence interval for each prediction, from which the model can be learned adaptively and the exploration-exploitation tradeoff is handled in an informed way; (iii) the proposed algorithms outperform state-of-the-art competitors in both the theoretical and empirical results.

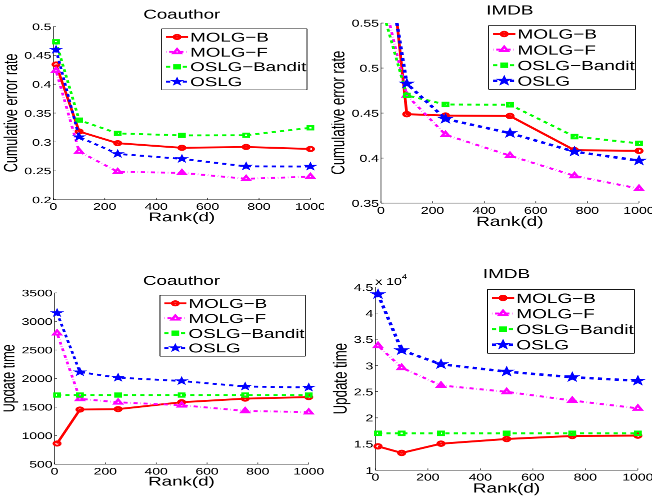

Figure 2: Sensitivity study of low rank impact on performance.

Appendix

Proof of Thm 2. Inspired by the proof in [Forster, 1999], we derive $\inf_{\mathbf{U}} G_t(\mathbf{U}) - \inf_{\mathbf{U}} G_{t-1}(\mathbf{U}) = \ell_t(alg) - a_t^2 \mathbf{x}_t^\top \mathbf{A}_t^{-1} \mathbf{x}_t + a_t - 1$. Summing over $t = 1, \ldots, T$ and expanding the square, $\sum_{t=1}^T (-\mathbf{f}_t \mathbf{y}_t - \sigma_t) \leq -\sum_{t=1}^T a_t \mathbf{y}_t \mathbf{U}^\top \mathbf{x}_t + \frac{1}{2}\mathrm{tr}(\mathbf{U}^\top \mathbf{A}_T \mathbf{U})$, where we assume $\|\mathbf{y}_t\|^2 = 1$ and $\mathbf{A}_T = b\mathbf{I} + \sum_{t=1}^T a_t \mathbf{x}_t \mathbf{x}_t^\top$. We add $\sum_t a_t$ on both sides and $a_t = \frac{1}{1 - \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1} \mathbf{x}_t} > 1$, $\sum_t a_t \tilde{\mathcal{L}}(\mathbf{y}_t \cdot \mathbf{U}^\top \mathbf{x}_t) + \frac{1}{2}\mathrm{tr}(\mathbf{U}^\top \mathbf{A}_T \mathbf{U}) \geq \sum_t (1 - \mathbf{f}_t \mathbf{y}_t - \sigma_t)$, where the hinge loss $\tilde{\mathcal{L}}(x) = \max(0, 1 - x) \geq 1 - x$. Two types of update trials: (I) when an error occurs, i.e., $t \in \mathcal{M}$ and $-\mathbf{f}_t \mathbf{y}_t \geq 0$, $\sum_t (1 - \mathbf{f}_t \mathbf{y}_t - \sigma_t) \geq M - \sum_{t \in \mathcal{M}} \sigma_t$; (I-I) when no error occurs, i.e. $t \in \mathcal{D}$ and $0 \leq \mathbf{f}_t \mathbf{y}_t \leq \sigma_t$, $\sum_t (1 - \mathbf{f}_t \mathbf{y}_t + \sigma_t - 2\sigma_t) \geq D - 2\sum_{t \in \mathcal{D}} \sigma_t$. Combine two cases with $\sum_{t=1}^T \sigma_t \leq \frac{b}{2(b-1)} \ln(\frac{1}{b} \mathbf{A}_T)$, we finish the proof. □

References

[Abernethy *et al.*, 2009] Jacob Abernethy, Alekh Agarwal, and Peter L Bartlett. A stochastic view of optimal regret through minimax duality. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

[Agarwal, 2013] Alekh Agarwal. Selective sampling algorithms for cost-sensitive multiclass prediction. In *ICML-13*, pages 1220–1228, 2013.

[Auer, 2002] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *JMLR*, 3(Nov):397–422, 2002.

[Azoury and Warmuth, 2001] Katy S Azoury and Manfred K Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.

[Belkin *et al.*, 2006] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7:2399–2434, 2006.

[Beygelzimer *et al.*, 2017] Alina Beygelzimer, Francesco Orabona, and Chicheng Zhang. Efficient online bandit multiclass learning with $\tilde{O}(\sqrt{T})$ regret. In *ICML*, pages 488–497, 2017.

[Crammer and Gentile, 2013] Koby Crammer and Claudio Gentile. Multiclass classification with bandit feedback using adaptive regularization. *Machine learning*, 90(3):347–383, 2013.

[Dekel *et al.*, 2010] Ofer Dekel, Claudio Gentile, and Karthik Sridharan. Robust selective sampling from single and multiple teachers. In *COLT*, pages 346–358, 2010.

[Eckart and Young, 1936] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[Forster, 1999] Jürgen Forster. On relative loss bounds in generalized linear regression. In *Fundamentals of Computation Theory*, pages 269–280, 1999.

[Gu and Han, 2014] Quanquan Gu and Jiawei Han. Online spectral learning on a graph with bandit feedback. In *ICDM-14*, pages 833–838. IEEE, 2014.

[Gu *et al.*, 2013] Quanquan Gu, Charu Aggarwal, Jialu Liu, and Jiawei Han. Selective sampling on graphs for classification. In *Proceedings of the 19th ACM SIGKDD*, pages 131–139, 2013.

[Herbster and Pontil, 2006] Mark Herbster and Massimiliano Pontil. Prediction on a graph with a perceptron. In *NIPS-06*, pages 577–584, 2006.

[Kuznetsov and Mohri, 2016] Vitaly Kuznetsov and Mehryar Mohri. Time series prediction and online learning. In *Conference on Learning Theory*, pages 1190–1213, 2016.

[Moroshko and Crammer, 2014] Edward Moroshko and Koby Crammer. Weighted last-step min–max algorithm with improved sub-logarithmic regret. *Theoretical Computer Science*, 558:107–124, 2014.

[Moroshko *et al.*, 2015] Edward Moroshko, Nina Vaits, and Koby Crammer. Second-order non-stationary online learning for regression. *JMLR*, 16:1481–1517, 2015.

[Smola and Kondor, 2003] Alexander J Smola and Risi Kondor. Kernels and regularization on graphs. In *Learning theory and kernel machines*, pages 144–158. Springer, 2003.

[Takimoto and Warmuth, 2000] Eiji Takimoto and Manfred K Warmuth. The last-step minimax algorithm. In *Algorithmic Learning Theory*, pages 279–290, 2000.

[Vovk, 1998] Volodya Vovk. Competitive on-line linear regression. *NIPS-98*, pages 364–370, 1998.

[Wang *et al.*, 2012] Jialei Wang, Peilin Zhao, and Steven C Hoi. Exact soft confidence-weighted learning. In *ICML*, pages 121–128, 2012.

[Yang *et al.*, 2016] Peng Yang, Peilin Zhao, Zhen Hai, Wei Liu, Steven CH Hoi, and Xiao-Li Li. Efficient multi-class selective sampling on graphs. In *UAI*, page 805. AUAI Press, 2016.