Bandit Online Learning on Graphs via Adaptive Optimization

Peng Yang[1], **Peilin Zhao**[2,3] and **Xin Gao**[1]
[1]King Abdullah University of Science and Technology, Saudi Arabia
[2]South China University of Technology, China
[3]Tencent AI Lab, China
{peng.yang.2, xin.gao}@kaust.edu.sa; peilinzhao@hotmail.com

Abstract

Traditional online learning on graphs adapts graph Laplacian into ridge regression, which may not guarantee reasonable accuracy when the data are adversarially generated. To solve this issue, we exploit an adaptive optimization framework for online classification on graphs. The derived model can achieve a min-max regret under an adversarial mechanism 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. However, this algorithm assumes that the full information label is provided for each node, which is violated in many practical applications where labeling is expensive and the oracle may only tell whether the prediction is correct or not. To address this issue, we propose a bandit online algorithm on graphs. It derives per-instance confidence region of the prediction, from which the model can be learned adaptively to minimize the online regret. 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 learning on graphs usually adapts the graph Lapalacian regularization into ridge regression [Herbster and Pontil, 2006; Gu *et al.*, 2013], which may not guarantee sufficiently high performance in a non-stationary environment where the label for each node (i.e. $\mathbf{x}_t$) is adversarially generated [Vovk, 1998]. To solve this problem, online learning has been extensively studied in the adversarial setting [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 of the prediction of a given node, the learner receives a binary feedback indicating whether the predicted label is correct or not. An early work of this problem is Banditron [Kakade *et al.*, 2008], a first-order bandit online algorithm. However, this model cannot optimize the direction of update. To solve this issue, a second-order bandit algorithm was proposed, which tracks a spectral structure of observed inputs [Crammer and Gentile, 2013]. However, previous algorithms were seldom designed on graphs, among which Gu et al. [Gu and Han, 2014] derived a bandit online algorithm with Laplacian regularization. Although this method is effective in solving the bandit problem, it is limited by a fix-margin rule for model learning, instead of an adaptive learning strategy [Yang *et al.*, 2015]. Recent studies focus on the general contextual bandit problem on graphs [Carpentier and Valko, 2016; Alon *et al.*, 2015], where context-reward pairs are generated i.i.d., which we do not assume here. For a comprehensive survey on Online Learning, please refer to [Hoi *et al.*, 2018].

In this paper, based on the observation above, we present an adaptive optimization framework for online classification on graphs. 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 algorithms using error-driven update rules. However, this algorithm assumes that the true label is provided for every node, which is impractical in many scenarios where labeling is expensive and label oracle may only give a one-bit feedback indicating whether the prediction is correct. To solve this issue, we propose a bandit online algorithm on graphs. We adapt both confident weight and adaptive margin into bandit setting, with which the model can be learned adaptively to significantly reduce the regret of bandit online learning. Experiments on several benchmark graph datasets show that the proposed bandit algorithm outperforms several competitors, even sometimes beats state-of-the-art algorithms that use full label feedback.

Notation. With an appropriate size, an identity matrix is denoted as $\mathbf{I}$, zero vector as $\mathbf{0}$ and diagonal matrix as $\text{diag}(\cdot)$. 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 a 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 $\text{tr}(\mathbf{A})$ 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 the graph Laplacian is $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$, where $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ are the eigenvectors, and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_n)$ $(0 = \lambda_1 \leq \ldots \leq \lambda_n \leq 2)$ are the eigenvalues.

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

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

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 the graph, this optimization is to learn the function $\mathbf{F}$ that satisfies two requirements: (i) the predictions for labeled nodes should be close to the given labels for those nodes; (ii) label-smoothness on graphs: the nodes nearby on graphs should have similar predictions.

To solve 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 the 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 $\mathbf{X}\mathbf{L}\mathbf{X}^\top = \mathbf{I}$, 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 the ridge regression formulation. Eq. (3) can be used to derive the online learning under a new graph node representation.

2.1 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 $\mathbf{F}$.

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., $\text{tr}(\tilde{\mathbf{F}}^\top \mathbf{L} \tilde{\mathbf{F}}) = \|\tilde{\mathbf{W}}\|_F^2$, while the graph regularized problem can be formulated as a low-rank form, i.e., $\inf_{\tilde{\mathbf{W}}} \sum_{i=1}^n \ell(\tilde{\mathbf{W}}; (\tilde{\mathbf{x}}_i, \mathbf{y}_i)) + b\|\tilde{\mathbf{W}}\|_F^2$.

3 Online Learning on Graphs

The objective of online learning on graphs aims to achieve a low regret compared to 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, an 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 the model and then proceeds to the next round. For any matrix $\mathbf{U} \in \mathbb{R}^{d \times K}$, 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

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

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 Adaptive Online Optimization

In the adversarial environment, the regret is solved based on a last-step min-max optimization [Forster, 1999; Yang and Zhao, 2015]. 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, respectively, 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; Yang and Zhao, 2015], 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, our variant of the adaptive-weighted algorithm predicts,

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

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

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

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 weighted problem could be concave in y_T and convex in $\hat{y}_T$. The optimal solution of our objective function 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 adaptive-weighted function $\hat{y}_T = \arg\min_{\hat{y}_T} \max_{y_T} F(\mathbf{y}_T, \mathbf{f}_T)$ is*

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

Proof. The proof is provided on the website[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 the paper. The a_T depends on previous learned rounds $\{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ with a fixed computational cost $O(d^2)$.

Specifically, the above model maintains two quantities: $\mathbf{A}_t$ and $\mathbf{B}_t$. Inspired by [Azoury and Warmuth, 2001], we exploit the 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 the 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 $O(d^2)$ and the required memory is $O(d^2)$.

Assume our algorithm updates the model when an error occurs ($\mathbf{f}_t \cdot \mathbf{y}_t \leq 0$), then its mistake bound holds in the following lemma.

Theorem 2 *Assume $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)\}$ is a node sequence, an online algorithm predicts with $\hat{y}_T = \arg\max_{i \in [K]} (\mathbf{B}_{T-1}^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T)_i$, and updates $\mathbf{A}_T$ and $\mathbf{B}_T$ with Eq. (6). Let $\mathcal{Z} = \{t \mid \mathbf{f}_t \cdot \mathbf{y}_t \leq 0\}$ be updated trials, then the following inequality holds for any $\mathbf{U} \in \mathbb{R}^{d \times K}$,*

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

Proof. The proof is provided on the website[1]. □

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

```
Input: Adjacency matrix $\mathbf{S}$, rank $d$, and regularization parameters $b$ and $\phi$.
Output: $\mathbf{W}_T$.
Initialize: $\mathbf{W}_0 = \mathbf{0}, \mathbf{B}_0 = \mathbf{0}, \mathbf{A}_0 = b\mathbf{I}$.
Compute $\mathbf{L} = \mathbf{D} - \mathbf{S}$ and $\mathbf{X}$ from $\mathbf{L}^\dagger$;
for $t = 1, \ldots, T$ do
    Let $Z_t = 0$; Receive $\mathbf{x}_t$;
    $\mathbf{A}_t^{-1} = (\mathbf{A}_{t-1} + a_t \mathbf{x}_t \mathbf{x}_t^\top)^{-1}$, $\mathbf{W}_{t-1} = \mathbf{A}_t^{-1} \mathbf{B}_{t-1}$;
    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
```

3.2 Adaptive-Margin Learning

When our algorithm uses an error-driven update rule (e.g., $\mathbf{f}_t \cdot \mathbf{y}_t \leq 0$), it usually ignores the correctly predicted labels of low predicted confidence, especially in early stage of online learning. To solve this issue, we propose an adaptive large-margin update rule. We begin with an annotation of the prediction margin in multiclass classification,

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

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 the model whenever the function $\Theta_t < 0$, where Θ_t is an adaptive margin towards the current prediction,*

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

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 defined in Eq. (8).

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 current prediction. In this way, $\Theta_t = \hat{\Delta}_t - \sigma_t$ acts as the *lower confidence bound* of the prediction. Note that the adaptive-margin conception has been studied in [Wang *et al.*, 2012; Yang *et al.*, 2016]. Different from the previous methods, our Θ_t is derived from the weighted min-max framework on graphs. We summarize the algorithm, the Adaptive Optimization for Online Learning on Graph (defined by (6), (7) and (9)), in Algorithm 1. We next theoretically analyze the effectiveness of our algorithm.

In Algorithm 1, the update trials are 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 an update is issued when an error occurs, and $\mathcal{D} = \{t : 0 < \hat{\Delta}_t < \sigma_t, \hat{y}_t = y_t\}$ with

[1] https://github.com/YoungBigBird1985/MOLG/

$D = |\mathcal{D}|$ 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 the update trials containing $Z = M + D$.

Theorem 3 *Algorithm 1 runs on an arbitrary node sequence* $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)\}$. *Let* $\tilde{\mathcal{L}}(x) = \max(0, 1-x)$ *be hinge loss, the following inequality holds for any* $\mathbf{U} \in \mathbb{R}^{d\times K}$,

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

Proof. The proof is provided on our website[1]. □

Remark. Since $\mathbf{A}_t^{-1} \preceq \mathbf{A}_{t-1}^{-1}$ is decreased over t, we have $\mathbf{x}_t^\top \mathbf{A}_t^{-1} \mathbf{x}_t \leq \ldots \leq \mathbf{x}_t^\top \mathbf{A}_0^{-1} \mathbf{x}_t \leq \frac{1}{b}$ where $b > 1$ and $\|\mathbf{x}\|_2 \leq 1$. Thus, $a_t = \frac{1}{1-\mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1} \mathbf{x}_t} \leq \frac{b}{b-1}$, and $\text{tr}(\mathbf{U}^\top \mathbf{A}_\mathcal{Z} \mathbf{U}) \leq \sum_{t\in\mathcal{Z}} a_t \|\mathbf{U}^\top \mathbf{x}_t\|^2 \leq \frac{Zb}{b-1} \|\mathbf{U}\|_F^2$. Moreover, $\log |\frac{1}{b} \mathbf{A}_T| \leq Kd \log(1+\frac{T}{Kdb})$ [Moroshko and Crammer, 2014]. Due to the deduction of the aggressive update trials $|\mathcal{D}|$, we claim that MOLG-F achieves a lower mistake bound than the algorithm using error-driven update rule in Theorem 2.

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 over the update trials,

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

In a low-rank setting where $d \ll T$, the above bound achieves a regret of $O(\log T)$ with respect to the best linear model.

3.3 Online Learning with Bandit Feedback

We propose a bandit online algorithm on graphs, MOLG-B, in Algorithm 2. Note that MOLG-B is the first work that adapts both confidence weight and adaptive margin into bandit setting. Specifically, confident weight a_t for the parameter update can guide the magnitude of bandit learning. Moreover, different from the update rules in Confidit [Crammer and Gentile, 2013] or SOBA [Beygelzimer *et al.*, 2017], MOLG-B proposes an adaptive-margin update strategy, which updates model based on the confidence of current prediction. These techniques improve the efficiency of bandit online learning.

In bandit setting, MOLG-B receives a node $\mathbf{x}_t$ at round t and predicts its label as $\hat{y}_t \in [K]$. Different from Algorithm 1, the learner receives a 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 $\sum_{i=1}^K \Delta_t^i = 2 - K$.

MOLG-B maintains a set of model parameters for multiple classes $\{(\mathbf{b}^i \in \mathbb{R}^d, \mathbf{A}^i \in \mathbb{R}^{d\times d})\}_{i=1}^K$, initialized with $\mathbf{b}_0 = \mathbf{0}, \mathbf{A}_0 = b\mathbf{I}$. Provided with 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 the current prediction $\hat{\Delta}_t^i$. The bandit algorithm chooses an estimation from the prediction margin and the predicted confidence, so that the upper confidence bound (UCB) of the prediction 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 UCB technique has been widely used in bandit online learning [Crammer and Gentile, 2013; Gu and Han, 2014; Beygelzimer *et al.*, 2017].

Algorithm 2 MOLG-B: Adaptive Optimization for Online Learning on Graphs with Bandit Feedback

```
Input: Adjacency matrix $\mathbf{S}$, rank $d$, and regularization parameters $b$, $\varphi$ and $\phi$.
Output: $\mathbf{W}_T$.
Initialize: $\mathbf{b}_0^i = \mathbf{0}$, $\mathbf{A}_0^i = b\mathbf{I}$ for all $i \in [K]$.
Compute $\mathbf{L} = \mathbf{D} - \mathbf{S}$ and $\mathbf{X}$ from $\mathbf{L}^\dagger$;
for $t = 1, \ldots, T$ do
    Let $Z_t = 0$; 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, $R_T = \sum_{t=1}^T (\mathbb{P}_t(y_t \neq \hat{y}_t) - \mathbb{P}_t(y_t \neq y_t^*)) = \sum_{t=1}^T \frac{\Delta_t - \hat{\Delta}_t}{2}$, where $\mathbb{P}_t = \mathbb{P}(\cdot|\mathbf{x}_1, \ldots, \mathbf{x}_t, y_1, \ldots, y_t)$ is the conditional probability.

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

$$R_T \leq \sqrt{(\frac{b}{b-1})^3 T} (\sqrt{H_1 H_2} + H_2),$$

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* $H_2 = 2Kd\gamma\log(1+\frac{T}{Kdb})$.

Proof. The proof is provided on the website[1]. □

Remark. Theorem 4 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., $O(\log T)$. However, it is worth noting that the two regrets are incomparable due to the different regret definitions in two algorithms.

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 datasets to evaluate all the algorithms: 1) Coauthor[2] is a coauthor graph of the *DBLP* dataset, with 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[2] 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[3] is an up-to-date movie dataset with 17046 motives as nodes and 993528 co-actor relations as edges. Each movie is labeled by one of 4 genres: “Animation”, “Action”, “Thriller” and “Romance”. 4) PubMed[4] is a graph of 19717 scientific publications of diabetes labeled by one of 3 types. It has 88651 citation links.

All graphs are supposed to be undirected and connected. In case some edges are directed, they are transformed into undirected ones. If a graph is 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, and (ii) cumulative number of updates. The error rate reflects the prediction accuracy of the online learning while the number of updates measures the computational efficiency of an online algorithm. Note that the smaller the two measures, the better the performance of an online algorithm. In order to fairly compare the algorithms, 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 graphs. (b) *OSLG* [Gu and Han, 2014] and *MSG* [Yang *et al.*, 2016] are second-order online algorithms on graphs with full information 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 recommended instructions. (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 structures. 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 of MOLG and OSLG over GPA is always significant on the four datasets. This is consistent with previous observations in online learning: the second-order algorithms are generally better than the 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 smaller number of updates and lower error rates than OSLG-Bandit and Banditron. The possible reasons are two folds: (1) the confidence weight can guide the magnitude and direction of the parameter update; (2) the adaptive-margin update rule can prioritize the instances with predicted uncertainty. These techniques speed up the convergence of online learning, so that the number of update and the error rate can be reduced further when the model has learned sufficient knowledge of the data. It demonstrates both the computational efficiency and label efficiency of our algorithm.

MOLG-B is slightly worse than MOLG-F in Coauthor and Cora datasets. This is reasonable, since MOLG-B uses partial information labels, whereas MOLG-F uses the full information labels. Thus, the performance of the bandit algorithm should be no better than that of the algorithm in the full information setting. However, MOLG-F requires more updates than MOLG-B to achieve better accuracy. The reason is obvious: provided with 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 the parameters of multiple classes. Overall, MOLG-B is computationally more efficient than MOLG-F.

Finally, MOLG-B outperforms MOLG-F in IMDB and PubMed datasets. The reason may be due to class-imbalanced issue. In the multiclass setting, for each class, its training instances are fewer than that of negative classes (remaining $K-1$ classes). MOLG-B updates one class model per trial, while MOLG-F can exploit full label information to update the models of multiple classes in each round. Thus, in MOLG-F, negative update may dominate the predictive model and lead to poor performance [Zhang *et al.*, 2016].

4.3 Sensitivity Analysis on Update Ratio

In the adaptive-margin update rule, the tradeoff parameter ϕ has an influence on the update ratio. The model with a smaller value of ϕ would like to conduct a fewer number of update and vice versa. Specifically, we set the ϕ to $\{10^{-2}, \ldots, 10\}$

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

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

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

Algorithm	*Coauthor*			*Cora*	
	Error rate (%)	# Update number		Error rate (%)	# Update number
GPA	54.74±2.66	3422		58.49±2.03	4970
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 (%)	# Update number		Error rate (%)	# Update number
GPA	68.70±0.46	34092		57.95±0.29	39434
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**

Table 1: Performance in terms of mean and standard deviation (Banditron, OSLG-Bandit and GPA update all the rounds. Bandit algorithms choose one class to update its parameter in each round. Banditron represents the best result from Confidit and SOBA)

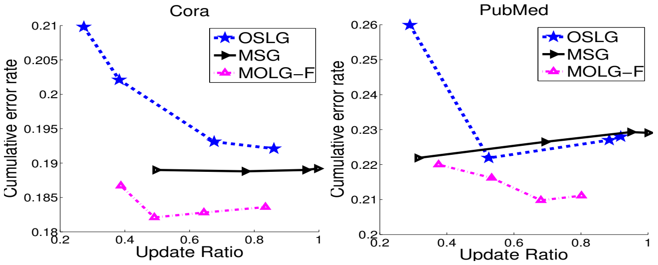

Figure 1: Sensitivity analysis of the update ratio.

and calculate the averaged ratio of update over the 20 times of experiments. MSG, OSLG and MOLG-F are used for comparison since other baselines update all the rounds. The result in Fig.1 shows that MOLG-F achieves the best performance consistently under different ratios of update. OSLG uses an error-driven update rule, thus ignoring the correctly predicted labels. The better empirical performance achieved by MOLG-F shows that the labels with low predicted confidence are informative to improve the performance of online model.

4.4 Sensitivity Analysis of Low-rank Graph Data

The empirical results are 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 the case studies 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 approximations. Note that, in the bandit setting, a higher rank increases the uncertainty of the data, thus the bandit algorithms require more updates to achieve a lower or comparable error rate. In the full information setting, a higher rank increases the knowledge of the data, thus the updates can be further reduced when the model learns sufficient information. However, with a higher rank, the algorithms need higher computational time to update the model. To achieve a balance, we choose $d = 100$ since the algorithm achieves a good accuracy while the number of update is small enough.

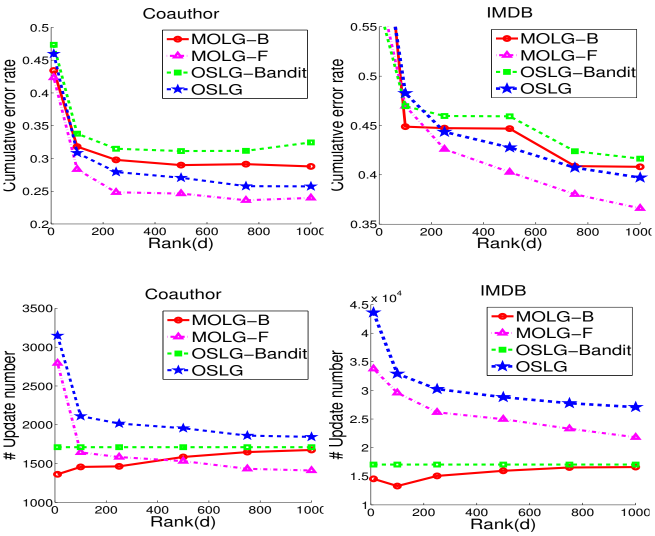

Figure 2: Sensitivity analysis of low rank impact on performance.

5 Conclusion and Future Work

We explore adaptive optimization for online graph classification with full information and bandit label feedbacks. The main contributions of our work are four folds: (i) We propose a weighted min-max optimization framework for online learning on graphs; (ii) An adaptive-margin update rule is proposed, which achieves a lower regret than the algorithm using error-driven update rule; (iii) We adapt both confidence weight and adaptive margin into bandit setting, which significantly reduces the regret of bandit learning; (iv) the proposed algorithms outperform state-of-the-art competitors in the empirical results. In the future, we shall study several other directions for online learning on Graphs, including: online optimization of ranking performance [Zhao *et al.*, 2011], multi-task online learning [Yang *et al.*, 2017], etc.

References

[Abernethy *et al.*, 2009] Jacob Abernethy, Alekh Agarwal, and Peter L Bartlett. A stochastic view of optimal regret through minimax duality. In *COLT*, 2009.

[Alon *et al.*, 2015] Noga Alon, Nicolo Cesa-Bianchi, Ofer Dekel, and Tomer Koren. Online learning with feedback graphs: Beyond bandits. In *COLT*, pages 23–35, 2015.

[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.

[Carpentier and Valko, 2016] Alexandra Carpentier and Michal Valko. Revealing graph bandits for maximizing local influence. In *AISTATS*, pages 10–18, 2016.

[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.

[Hoi *et al.*, 2018] Steven C. H. Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey. *CoRR*, abs/1802.02871, 2018.

[Kakade *et al.*, 2008] Sham M Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Efficient bandit algorithms for online multiclass prediction. In *ICML*, pages 440–447. ACM, 2008.

[Kuznetsov and Mohri, 2016] Vitaly Kuznetsov and Mehryar Mohri. Time series prediction and online learning. In *COLT*, 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.

[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 and Zhao, 2015] Peng Yang and Peilin Zhao. A minmax optimization framework for online graph classification. In *CIKM*, pages 643–652. ACM, 2015.

[Yang *et al.*, 2015] Peng Yang, Peilin Zhao, Vincent W Zheng, and Xiao-Li Li. An aggressive graph-based selective sampling algorithm for classification. In *ICDM*, pages 509–518. IEEE, 2015.

[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.

[Yang *et al.*, 2017] Peng Yang, Peilin Zhao, and Xin Gao. Robust online multi-task learning with correlative and personalized structures. *IEEE Trans. Knowl. Data Eng.*, 29(11):2510–2521, 2017.

[Zhang *et al.*, 2016] Xiaoxuan Zhang, Tianbao Yang, and Padmini Srinivasan. Online asymmetric active learning with imbalanced data. In *ACM SIGKDD*, pages 2055–2064. ACM, 2016.

[Zhao *et al.*, 2011] Peilin Zhao, Steven C. H. Hoi, Rong Jin, and Tianbao Yang. Online AUC maximization. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 233–240, 2011.