Efficient Multi-Class Selective Sampling on Graphs

Peng Yang[‡] and **Peilin Zhao**[‡] and **Zhen Hai** [‡] and **Wei Liu**[†] and **Steven C.H. Hoi** [♯] and **Xiao-Li Li**[‡]
[‡] Institute for Infocomm Research, Singapore, 138632. Email: {yangp,zhaop,haiz,xlli}@i2r.a-star.edu.sg
[†] Tencent AI Lab, Shenzhen, China, 518057. Email: wliu@ee.columbia.edu
[♯] School of Information Systems, Singapore Management University, Singapore, 178902. Email: chhoi@smu.edu.sg

Abstract

A graph-based multi-class classification problem is typically converted into a collection of binary classification tasks via the one-vs.-all strategy, and then tackled by applying proper binary classification algorithms. Unlike the one-vs.-all strategy, we suggest a unified framework which operates directly on the multi-class problem without reducing it to a collection of binary tasks. Moreover, this framework makes active learning practically feasible for multi-class problems, while the one-vs.-all strategy cannot. Specifically, we employ a novel randomized query technique to prioritize the informative instances. This query technique based on the hybrid criterion of "margin" and "uncertainty" can achieve a comparable mistake bound with its fully supervised counterpart. To take full advantage of correctly predicted labels discarded in traditional conservative algorithms, we propose an aggressive selective sampling algorithm that can update the model even if no error occurs. Thanks to the aggressive updating strategy, the aggressive algorithm attains a lower mistake bound than its conservative competitors in expectation. Encouraging experimental results on real-world graph databases show that the proposed technique by querying an extremely small ratio of labels is able to accomplish better classification accuracy.

I. INTRODUCTION

Graphs, as a family of ubiquitous structures to model different types of networks, such as social networks (*e.g.*, Facebook, Twitter), biological networks [19], [20], and citation networks [21], have been widely applied in diverse applications. Particularly, one important task is to classify graph vertices into multiple classes, *e.g.*, authors in a citation network can be classified into different domains/classes, such as computer science, biology, physics, mathematics, economics, *etc*. To build a classifier, the desired classification model can be learnt from a set of vertex-label pairs in both offline [7] and online settings [16]. Offline algorithms can access the labels of all the stored vertices in a pool, which increases the storage requirement. Online learning, on the other hand, obtains the instances in a sequential order. It allows to access the label of the current vertex; after updating the model, the current input will be discarded [18]. Therefore, online learning is scalable to deal with massive datasets.

Although online classification on graphs has been well studied, it still remains as a challenging research subject, which is primarily due to three reasons. First, most online techniques focus on binary classification problems. Some approaches [14], [22] address multi-class problems by using output coding [12]. Such a setting may be ineffective and ill-defined since it generalizes multiple binary classifiers and each classifier is maintained and updated *independently* of the others [1]. Second, online learning assumes that the labels of all vertices are provided already. It is impractical as labeling every sample is expensive and time-consuming in many real-world applications. Third, in social networks, data usually arrives in a sequential order and the network scale can be very large, which brings a critical challenge to develop efficient and scalable algorithms for graph classification.

To address the aforementioned challenges, we present a unified framework to cope with multi-class online classification on graphs. Specifically, we adapt the graph Laplacian Regularized Least Squares (LapRLS) model to the multi-class setting, in which updating one class model has a global impact on the other classes. However, such an approach assumes that all labels are available, which obviously limits its usage to many domains. To minimize the labeling cost, we propose a new query technique based on both the "margin" [9] and "uncertainty" criteria, to only query the labels of the most informative instances. We

[1]Refer to the comparison between the binary and multi-class settings in the supplementary material.

theoretically analyze an online algorithm running on the selected labels by our query technique, which achieves a comparable mistake bound with the one that queries all labels. In addition, to take full advantage of correctly predicted labels that are discarded in conservative algorithms, we introduce an aggressive version of selective sampling. It hybrids the conservative update with the aggressive sampling scheme, which updates the model even if no error occurs. The theoretical results show that our aggressive selective sampling algorithm can achieve better performance than its conservative competitors. Extensive experiments carried out on several real-world graph datasets further validate the empirical performance of the proposed algorithms.

The rest of this paper is organized as follows. Section 2 presents the problem setting of graph classification. The proposed multi-class online learning and selective sampling algorithms are described in Section 3 and Section 4, respectively. Section 5 discusses the experimental results. Section 6 concludes our work.

II. GRAPH CLASSIFICATION

In this section, we first present the notations. Then we introduce a graph Laplacian regularization that can derive a linear model for multi-class classification.

A. Notation

In this paper, we will use lower case letters as scalars (e.g. x), lower case bold letters as vectors (e.g. $\mathbf{f}$), upper case letters as elements of a matrix (e.g. S_{ij}) and boldface upper letters as matrices (e.g. $\mathbf{S}$). With an appropriate size, an identity matrix is defined as $\mathbf{I}$ and a vector of all zeros as $\mathbf{0}$. The transpose of a vector $\mathbf{m}$ is denoted as $\mathbf{m}^\top$, the inverse of a matrix $\mathbf{A}$ as $\mathbf{A}^{-1}$, and the pseudo inverse of $\mathbf{A}$ as $\mathbf{A}^\dagger$. A diagonal matrix is denoted as $\text{diag}(\sigma_1, \ldots, \sigma_n)$ with diagonal elements $\sigma_i, i \in [1, n]$. In addition, Euclidean norms are denoted as $\|\cdot\|_2$, Frobenius norm as $\|\cdot\|_F$ and the trace of square matrix as $tr(\cdot)$. When function $f(W)$ is differentiable, we denote its gradient by $\nabla f(W)$.

We consider the problem of classification in probabilistic setting: n i.i.d. pais are generated by a probability distribution on $\mathcal{X} \times \mathcal{Y}$, where y_i in a pair of (x_i, y_i) is the $class$ of instance x_i. We define $|\mathcal{Y}| = 2$ as binary-class setting, and $|\mathcal{Y}| > 2$ as a multi-class problem.

B. Graph Laplacian Regularization

$G = (V, E)$ is defined as a graph with an vertex set $V = \{v_1, \ldots, v_n\}$, an edge set $E = \{(v_i, v_j) | v_i, v_j \in V\}$ and an adjacency matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$, where the element $S_{ij} \in \mathbb{R}_0^+$ is measured by the affinity of edge (v_i, v_j). We assume graph G is connected and undirected in this work. Given D is the diagonal matrix with $D_{ii} = \sum_j S_{ij}$, graph Laplacian is defined as $L = D - S$ with its eigenvector $\mathbb{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n] (\mathbf{v}_i \in \mathbb{R}^n)$ and eigenvalue $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ where $0 = \lambda_1 \leq \ldots \leq \lambda_n$. Intuitively, the objective function incurs a heavy penalty, if neighboring vertices v_i and v_j are mapped far apart. The graph regularization [17] assumes a label smoothness over the graph,

$$\frac{1}{2} \sum_{i,j=1}^{n} S_{ij} \|\mathbf{f}_i - \mathbf{f}_j\|^2 = tr\left(F^\top (D - S) F\right) = tr\left(F^\top L F\right),$$

where $F = [\mathbf{f}_1, \ldots, \mathbf{f}_n]^\top$ and $\mathbf{f}_i \in \mathbb{R}^K$ is the prediction scores of node i on K classes.

In the setting of graph classification, the real-valued function satisfies: 1) the values of function F for labeled vertices should be close to the given labels for that vertices; 2) vertices should satisfy label smoothness on the whole graph, that is, the points nearby in graph should have similar labels. In the multi-class scenario, the generalized LapRLS solves the following function,

$$\min_F \|F - Y\|_F^2 + \gamma tr\left(F^\top L F\right), \tag{1}$$

where $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^\top \in \mathbb{R}^{n \times K}$, $\mathbf{y}_i \in \mathbb{R}^K$ is the true label of node i on K classes, and $\gamma > 0$ is a regularized parameter for graph Laplacian. To solve the problem (1) with a linear model, we consider its dual form. Using the definition of graph kernel [17], the function F can be defined, $F = L^\dagger \Phi$, where $L^\dagger$ is the pseudo inverse of L and $\Phi \in \mathbb{R}^{n \times K}$ is a parameter matrix. Assuming that $L^\dagger = M^\top M = \sum_i \frac{1}{\lambda_i} \mathbf{v}_i \mathbf{v}_i^\top$, where $M = [\frac{1}{\sqrt{\lambda_1}} \mathbf{v}_1, \ldots, \frac{1}{\sqrt{\lambda_n}} \mathbf{v}_n]^\top \in \mathbb{R}^{n \times n}$ and $W = M\Phi \in \mathbb{R}^{n \times K}$, the kernel function F can be reformatted in a linear model form,

$$F = M^\top W, \tag{2}$$

where $\mathbf{f}_i = W^\top \mathbf{m}_i$ is essentially a linear regression model. In the multi-class setting, our linear model and its prediction margin are defined as follows.

Definition 1. *We define a multi-class linear model that consists of label score of an input $\mathbf{m}_i$ over the K classes,*

$$\mathbf{f}_i = \left[P_W^{\mathbf{m}_i}(1), \ldots, P_W^{\mathbf{m}_i}(K)\right]^\top. \tag{3}$$

Given the node-label pair $(\mathbf{m}_i, y_i)$, we define the prediction margin of $\mathbf{f}_i$,

$$P_W^{\mathbf{m}_i}(y_i) - \max_{j \neq y_i} P_W^{\mathbf{m}_i}(j) = \mathbf{f}_i \cdot \mathbf{y}_i, \tag{4}$$

where $\mathbf{y}_i \in \mathbb{R}^K$ is a label vector of node i with an entry of $+1$ for true class y_i, -1 for the class $j = \text{argmax}_{j \neq y_i} P_W^{\mathbf{m}_i}(j)$ and 0 otherwise.

Substituting Eq. (2) into Eq. (1), we obtain,

$$\min_W \|M^\top W - Y\|_F^2 + \gamma tr\left(W^\top W\right).$$

In this way, we derive a formulation, similar to ridge-regression, with an additional regularization to shrink the prediction. It helps us to derive a multi-class online model with a new data representation for graph vertex.

C. Low-rank Approximation

Given $M = [\frac{1}{\sqrt{\lambda_1}}\mathbf{v}_1, \ldots, \frac{1}{\sqrt{\lambda_n}}\mathbf{v}_n]^\top$, the matrix W is updated with the time complexity $O(n^2)$, that is computationally expensive in large graph datasets. To make our algorithm scalable to big graphs, we propose a low-rank approximation $\hat{M}$ as follows,

$$\hat{M} = [\frac{1}{\sqrt{\lambda_1}}\mathbf{v}_1, \ldots, \frac{1}{\sqrt{\lambda_d}}\mathbf{v}_d]^\top \in \mathbb{R}^{d\times n},\ \hat{W} = \hat{M}\Phi \in \mathbb{R}^{d\times K},$$

where $d \ll n$ and the time complexity of our algorithm becomes $O(d^2) \ll O(n^2)$. We analyze the impact of such low-rank approximation on the function $\hat{F} = \hat{M}^\top\hat{W}$. We have $\hat{F} = \hat{L}^\dagger\Phi$, where $\hat{L}^\dagger = \hat{M}^\top\hat{M} = \sum_{i=1}^{d}\frac{1}{\lambda_i}\mathbf{v}_i\mathbf{v}_i^\top$. Given that $\frac{1}{\lambda_1} \geq \ldots \geq \frac{1}{\lambda_n}$, $\hat{L}^\dagger$ holds the d largest eigenvalues of $L^\dagger$. In this case, $\hat{L}^\dagger$ is the best rank-d approximation of $L^\dagger$ [13], and thus $\hat{F}$ is the best rank-d approximation of F. Equipped with $\hat{M}$ and $\hat{W}$ a low-rank objective function for multi-class online classification can be rewritten as follows,

$$\arg\min_{\hat{W}} \sum_{t=1}^{T} \|\hat{W}^\top\hat{\mathbf{m}}_t - \mathbf{y}_t\|_2^2 + \gamma tr(\hat{W}^\top\hat{W}).$$

III. ONLINE LEARNING

Now we are ready to derive a multi-class online model on graph. We first present the problem setting of online learning. Then we derive the online classifier and its mistake bound.

A. Problem Setting

The purpose of online learning is to minimize the cumulative loss over the sequential nodes. Let $(\mathbf{m}_1, \mathbf{y}_1), \ldots, (\mathbf{m}_T, \mathbf{y}_T)$ $(T \leq n)$ be a sequence of vertices, where $\mathbf{m}_t \in \mathbb{R}^n$ is one column of matrix M and $\mathbf{y}_t \in \mathbb{R}^K$ is its label vector, an online version of LapRLS in multi-class setting is derived,

$$G_T(W) = \sum_{t=1}^{T} \|W^\top\mathbf{m}_t - \mathbf{y}_t\|_2^2 + \gamma tr\left(W^\top W\right). \quad (5)$$

At round t, online algorithm receives an input vertex $\mathbf{m}_t$, and predicts its label with the maximal score among the K classes, $\hat{y}_t = \underset{i\in[K]}{\operatorname{argmin}} P_{W_t}^{\mathbf{m}_t}(i)$. After prediction, its actual label y_t is revealed, and the algorithm uses it to update model and then proceeds to the next round. At each iteration, the performance of the online model is evaluated by a squared loss, $\ell_t(W) = \ell(y_t, \hat{y}_t) = \|\mathbf{y}_t - W^\top\mathbf{m}_t\|_2^2$ with cumulative loss over T iterations, $L_T(W) = \sum_{t=1}^{T}\ell_t(W)$. Similar, for any $U \in \mathbb{R}^{n\times K}$, let $\ell_t(U) = \|U^\top\mathbf{m}_t - \mathbf{y}_t\|_2^2$ be the instantaneous loss and $L_T(U) = \sum_{t=1}^{T}\ell_t(U)$ be the cumulative loss. The goal of online learning is to achieve low regret compared with the best linear function,

$$R_T = \sum_{t=1}^{T} g_t(W) - \inf_U \sum_{t=1}^{T} g_t(U),$$

where $G_T(W) = \sum_{t=1}^{T} g_t(W)$ and $g_t(W) = \ell_t(W) + \gamma tr\left(W^\top W\right)$ is regularized instantaneous loss on round t.

B. Online Learning on Graph

To minimize the regret, we have to minimize the cumulative loss $G_T(W)$ in the following lemma. We start with the notations,

$$A_T = \gamma I + \sum_{t=1}^{T}\mathbf{m}_t\mathbf{m}_t^\top, \quad B_T = \sum_{t=1}^{T}\mathbf{m}_t\mathbf{y}_t^\top. \quad (6)$$

Lemma 1. *For all $T \geq 1$, $G_T(W) = L_T(W) + \gamma tr(W^\top W)$ is minimal at an unique point W_T for all $T \geq 1$, given by*

$$W_T = A_T^{-1}B_T,\ G(W_T) = \sum_{t=1}^{T}\|\mathbf{y}_t\|^2 - tr\left(B_T^\top A_T^{-1}B_T\right).$$

We leave the proof in supplementary file. In Lemma 1, we obtain an optimal linear solution W_T. Inspired by [2], we exploit current input to predict its label with $\mathbf{f}_t = B_{t-1}^\top A_t^{-1}\mathbf{m}_t$ where $A_t = A_{t-1} + \mathbf{m}_t\mathbf{m}_t^\top$. However, it is not efficient to perform update in each iteration. To make it scalable on big graphs, we adopt a conservative strategy [4] to update model whenever an error occurs ($y_t \neq \hat{y}_t$). Note that our algorithm is different from [4], since the solution is a matrix for multi-class classification. We call our algorithm CMOG, a Conservative Multi-class Online learning on Graph, and summarize it in algorithm 1.

Although the CMOG is simple, it is the first work of online learning for solving graph-based multi-class problem. Below gives theoretical analysis of the CMOG and we begin with a lemma that facilitates the proof. With this lemma, we could then derive the mistake bound for the CMOG. For convenience, we introduce an additional notation:

$$r_t = \mathbf{m}_t^\top A_{t-1}^{-1}\mathbf{m}_t. \quad (7)$$

Then the following notation can be derived using Woodbury formula [3],

$$\begin{aligned}\mathbf{m}_t^\top A_t^{-1}\mathbf{m}_t &\overset{(6)}{=} \mathbf{m}_t^\top\left(A_{t-1} + \mathbf{m}_t\mathbf{m}_t^\top\right)^{-1}\mathbf{m}_t \\ &= \mathbf{m}_t^\top(A_{t-1}^{-1} - \frac{A_{t-1}^{-1}\mathbf{m}_t\mathbf{m}_t^\top A_{t-1}^{-1}}{1+\mathbf{m}_t^\top A_{t-1}^{-1}\mathbf{m}_t})\mathbf{m}_t = \frac{r_t}{1+r_t}.\end{aligned}$$

Lemma 2. *For all $t \geq 1$, $G_t(U)$ is the online LapRLS with any $U \in \mathbb{R}^{n\times K}$. Let $(\mathbf{m}_1, \mathbf{y}_1), \ldots, (\mathbf{m}_T, \mathbf{y}_T)$ be a sequence of input vertices, where $\mathbf{m}_t \in \mathbb{R}^n$ and $\mathbf{y}_t \in \mathbb{R}^K$, an online algorithm predicts with $\mathbf{f}_t = B_{t-1}^\top A_t^{-1}\mathbf{m}_t$. Then the following equality holds,*

$$\inf_U G_t(U) - \inf_U G_{t-1}(U) = \|\mathbf{y}_t - \mathbf{f}_t\|_2^2 - \frac{2r_t}{1+r_t} + r_t\|\mathbf{f}_t\|_2^2$$

We leave the proof in supplementary file. Based on the above lemma, we prove the following theorem that bounds the expected mistakes of CMOG. We denote

$\mathcal{M} = \{t|y_t \neq \hat{y}_t\}$ as the set of mistake trials with $|\mathcal{M}| = M$. For any model $U \in \mathbb{R}^{n\times K}$, let $\mathcal{U}_T$ be the set of its update trial, and $A_{\mathcal{U}_T} = \gamma I + \sum_{t\in\mathcal{U}_T} \mathbf{m}_t\mathbf{m}_t^\top$. Its hinge loss on round t is defined,

$$\mathcal{L}(\mathbf{y}_t^\top U^\top \mathbf{m}_t) = [1 - (P_U^{\mathbf{m}_t}(y_t) - \max_{j\neq y_t} P_U^{\mathbf{m}_t}(j))]_+.$$

Theorem 1. *Let $(\mathbf{m}_1, \mathbf{y}_1), \ldots, (\mathbf{m}_T, \mathbf{y}_T)$ be a sequence of inputs, where $\mathbf{m}_t \in \mathbb{R}^n$ and $\mathbf{y}_t \in \mathbb{R}^K$. Then for any model $U \in \mathbb{R}^{n\times K}$ and $h > 0$, the expected mistakes of CMOG (Alg. 1) on these sequential nodes is bounded by,*

$$\begin{aligned}\mathbb{E}[M] \leq& \mathbb{E}[\sum_t \mathcal{L}(\mathbf{y}_t^\top U^\top \mathbf{m}_t)] + \frac{h}{2} tr(U^\top \mathbb{E}[A_{\mathcal{U}_T}]U) \\ &+ \frac{1}{h}\mathbb{E}[\sum_t \frac{r_t}{1+r_t}].\end{aligned}$$

Remark 2. *$\sum_t \mathcal{L}(\mathbf{y}_t^\top U^\top \mathbf{m}_t)$ is the cumulative hinge loss made by U. Besides, for each class prototype $\mathbf{u}_i (i \in [K])$, $\mathbf{u}_i^\top A_{\mathcal{U}_T}\mathbf{u}_i$ lines between $\min_j \lambda_j$ and $\max_j \lambda_j$ where λ_j is an eigenvalue of the matrix $A_{\mathcal{U}_T}$. Thus, $tr(U^\top \mathbb{E}[A_{U_T}]U) \leq K \max_j \lambda_j$. Finally, $\sum_t \frac{r_t}{1+r_t} \leq \log \frac{\det(A_T)}{\det(A_0)} \leq n\log(R^2T+1)$ given $\|\mathbf{m}\|_2 \leq R$.*

Proof. The CMOG is a conservative algorithm that updates model whenever an error occurs. If there is no update, $U_t = U_{t-1}$ yields $\inf_U G_t(U) = \inf_U G_{t-1}(U)$. According to lemma 2, we have,

$$\begin{aligned}&\inf_U G_t(U) - \inf_U G_{t-1}(U) \\ =&\mathbb{I}\{y_t \neq \hat{y}_t\}(\|\mathbf{y}_t - \mathbf{f}_t\|_2^2 - \frac{2r_t}{1+r_t} + r_t\|\mathbf{f}_t\|_2^2)\end{aligned}$$

holds for all trial t. Summing over $t = 1, \ldots, T$, we obtain via expanding the squares and some manipulations,

$$\begin{aligned}&\sum_{t\in\mathcal{M}}(\|\mathbf{y}_t\|_2^2 - 2\mathbf{y}_t\cdot\mathbf{f}_t + \|\mathbf{f}_t\|_2^2 - \frac{2r_t}{1+r_t} + r_t\|\mathbf{f}_t\|_2^2) \\ =&\inf_U G_T(U) - \inf_U G_0(U) \\ \leq&\sum_{t\in\mathcal{M}}(\|\mathbf{y}_t\|_2^2 - 2\mathbf{y}_t\cdot U^\top\mathbf{m}_t) + tr(U^\top(\gamma I + \sum_{t\in\mathcal{U}_T}\mathbf{m}_t\mathbf{m}_t^\top)U)\end{aligned}$$

holding for any $U \in \mathbb{R}^{n\times K}$. We ignore $r_t\|\mathbf{f}_t\|_2^2$ as it does not affect upper bound. Given that $\hat{y} = \operatorname{argmax}_{j\neq y_t} P_{W_T}^{\mathbf{m}_t}(j)$,

$$\begin{aligned}1 - \mathbf{y}_t\cdot U^\top\mathbf{m}_t &\leq [1 - (P_U^{\mathbf{m}_t}(y_t) - P_U^{\mathbf{m}_t}(\hat{y}))]_+ \\ &\leq [1 - (P_U^{\mathbf{m}_t}(y_t) - \max_{j\neq y_i} P_U^{\mathbf{m}_t}(j))]_+ = \mathcal{L}(\mathbf{y}_t^\top U^\top \mathbf{m}_t).\end{aligned}$$

Since U is a random variable, we use hU $(h > 0)$ to replace U. We add $\sum_t 2h$ on both sides of inequality and simplify inequality with A_t and $\mathcal{L}(\cdot)$,

$$\begin{aligned}&\sum_{t\in\mathcal{M}}(\|\mathbf{f}_t\|_2^2 - 2\mathbf{y}_t\cdot\mathbf{f}_t - \frac{2r_t}{1+r_t} + 2h) \\ \leq& 2h\sum_{t\in\mathcal{M}}\mathcal{L}(\mathbf{y}_t^\top U^\top\mathbf{m}_t) + h^2 tr(U^\top A_{\mathcal{U}_T}U).\end{aligned} \tag{8}$$

Algorithm 1 CMOG: Conservative Multi-class Online model on Graph

Input: Adjacency matrix $\mathbf{S}$, and regularization parameter γ.
Output: W_T
Compute $\mathbf{L} = \mathbf{D} - \mathbf{S}$ and $\mathbf{M}$ from $\mathbf{L}$;
Initialize: $\mathbf{A}_0 = \gamma I$, $\mathbf{B}_0 = \mathbf{0}$, $W_0 = \mathbf{0}$;
for $t = 1, \ldots, T$ **do**
  Receive $\mathbf{m}_t \in R^n$;
  Compute $A_t^{-1} = (A_{t-1} + \mathbf{m}_t\mathbf{m}_t^\top)^{-1}$;
  Predict $\mathbf{f}_t = B_{t-1}^\top A_t^{-1}\mathbf{m}_t$;
  $\hat{y}_t = \arg\max_{j=1,..,K} \mathbf{f}_t(j)$;
  Query the actual label y_t;
  if $\hat{y}_t \neq y_t$ **then**
    Update $A_t = A_{t-1} + \mathbf{m}_t\mathbf{m}_t^\top$;
    Update $B_t = B_{t-1} + \mathbf{m}_t\mathbf{y}_t^\top$;
  else
    $A_t = A_{t-1}, B_t = B_{t-1}$;
  end if
end for
$W_T = A_T^{-1}B_T$;

When an error occurs (i.e., $y_t \neq \hat{y}_t$), we have that $P_{W_t}^{\mathbf{m}_t}(y_t) \leq P_{W_t}^{\mathbf{m}_t}(\hat{y}_t)$ yields $-\mathbf{y}_t\cdot\mathbf{f}_t = P_{W_t}^{\mathbf{m}_t}(\hat{y}_t) - P_{W_t}^{\mathbf{m}_t}(y_t) \geq 0$. With the expectation of the inequality, we bound given by $\|\mathbf{f}_t\|_2 > 0$,

$$\begin{aligned}&\sum_{t\in\mathcal{M}}\mathbb{E}[\|\mathbf{f}_t\|_2^2 - 2\mathbf{y}_t\cdot\mathbf{f}_t - \frac{2r_t}{1+r_t} + 2h] \\ \geq& 2h\mathbb{E}[M] - \mathbb{E}[\sum_t \frac{2r_t}{1+r_t}].\end{aligned} \tag{9}$$

Taking the expectation of Eq. (8) to upper bound the left-side of Eq. (9), we complete the proof. □

IV. SELECTIVE SAMPLING

In this section, we first introduce the setting of selective sampling. Next, we propose a novel randomized query approach to select labels and then introduce an aggressive algorithm that can use correctly predicted labels to optimize the model. We theoretically analyze mistake bound and query ratio of the proposed techniques. Finally, a low rank approximation is introduced in our framework.

A. Problem Setting

Unlike online algorithm that queries all labels, selective sampling has to decide whether to query label or not for each vertex $\mathbf{m}_t$. If a label $\mathbf{y}_t$ is queried of, the algorithm can update learner with $\mathbf{y}_t$; otherwise, no action is performed and the learner proceeds next one. Query and update decisions in trial t are denoted as binary variables Q_t and Z_t, respectively. When $Q_t = 1$ iif label $\mathbf{y}_t$ is queried of; $Q_t = 0$, no action performed. Update

decision Z_t is under similar setting. Generally, selective sampling is a semi-supervised online learning algorithm. Thus, its optimal solution can be derived in a form of online learning with query/update decision in each trial, i.e., $W_t = A_t^{-1}B_t$, where A_t and B_t can turn to be a recursive form,

$$A_t = A_{t-1} + Q_t Z_t \mathbf{m}_t \mathbf{m}_t^\top, \quad B_t = B_{t-1} + Q_t Z_t \mathbf{m}_t \mathbf{y}_t^\top.$$

Since A_t^{-1} is computationally expensive, we derive a non-inverted recursive form with time complexity $O(n^2)$ using Woodbury formula as in (8).

B. Label Query

The CMOG assumes that all labels are provided, which is not efficient in many real-world applications. To save the labeling cost, we propose a novel randomized query approach in multi-class setting. We begin with additional quantities of interest:

$$y_t^* = \underset{i=1,\dots,K}{\operatorname{argmax}} P_{U^*}^{\mathbf{m}_t}(i), \quad y_t^{'} = \underset{i \neq y_t^*}{\operatorname{argmax}} P_{U^*}^{\mathbf{m}_t}(i);$$
$$\hat{y}_t = \underset{i=1,\dots,K}{\operatorname{argmax}} P_{W_t}^{\mathbf{m}_t}(i), \quad y_t^{''} = \underset{i \neq \hat{y}_t}{\operatorname{argmax}} P_{W_t}^{\mathbf{m}_t}(i).$$

In words, y_t^* and $y_t^{'}$ are the optimal and second-best classes with respect to U^* (i.e., the best model in hindsight), while $\hat{y}_t$ and $y_t^{''}$ are the estimates of these classes based on our online learner W_t.

Definition 2. *Given an input $\mathbf{m}_t (t \in [T])$ and the weight $W_t = A_t^{-1}B_{t-1}$, an algorithm predicts its label with $\mathbf{f}_t = W_t^\top \mathbf{m}_t$, and queries the true label with a probability $\frac{2h}{2h+\max(0,\Theta_t)}$ ($h > 0$), where Θ_t is a confidence score towards current prediction,*

$$\Theta_t = \Theta(\mathbf{f}_t, r_t) = \frac{1}{2}\Delta_t^2 + 2\Delta_t - \frac{Kr_t}{1+r_t}, \qquad (10)$$

where $\Delta_t = P_{W_t}^{\mathbf{m}_t}(\hat{y}_t) - P_{W_t}^{\mathbf{m}_t}(y_t^{''})$.

This query is tuned by a confidence score Θ_t: a coin with bias $\frac{h}{h+\max(0,\Theta_t)}$ is flipped; if the coin turns up heads, then actual label $\mathbf{y}_t$ is queried; otherwise $Q_t = 0$ and no query performed. The randomized query has been studied in previous selective samplings under binary classification setting [6], [10]. Unlike these methods, we present a new confidence Θ_t based on the margin and uncertainty of the multi-class classification problems.

Intuitively, a query method is effective if it can control the probability of making a mistake whenever this label is not queried of. In the following theorem, we prove that an online algorithm on these selected labels $\{t|Q_t = 1, Q_t \sim \frac{2h}{2h+\max(0,\Theta_t)}\}$ can achieve a comparable mistake bound with one that queries all labels. Under randomized query, the mistake trials can be partitioned into two disjoint sets, $\mathcal{S} = \{t|\frac{2h}{2h+\max(0,\Theta_t)} < 1\}$ includes trials on which a stochastic query is conduct, while $\mathcal{D} = \{t|\frac{2h}{2h+\max(0,\Theta_t)} = 1\}$ includes trials when a deterministic query is issued.

Theorem 3. *For all $t \geq 1$, the CMOG runs over an arbitrary node-label sequence $(\mathbf{m}_1, \mathbf{y}_1), \dots, (\mathbf{m}_T, \mathbf{y}_T)$ ($\mathbf{m}_t \in \mathbb{R}^n$ and $\mathbf{y}_t \in \mathbb{R}^K$) with a query probability of $\frac{2h}{2h+\max(0,\Theta_t)}(h > 0)$ on round t, then the following inequality holds for any $U \in \mathbb{R}^{n\times K}$,*

$$\mathbb{E}[\mathcal{M}] \leq \mathbb{E}[\sum_t \mathcal{L}(\mathbf{y}_t^\top U^\top \mathbf{m}_t)] + \frac{h}{2} tr(U^\top \mathbb{E}[A_{\mathcal{U}_T}] U) + \frac{1}{2h}\mathbb{E}[\sum_{t\in\mathcal{M}\cap\mathcal{D}} \frac{Kr_t}{1+r_t}].$$

The expectation of queried number is upper bounded by $\mathbb{E}[|\mathcal{D}| + \sum_{t\in\mathcal{S}} \frac{2h}{2h+\Theta_t}]$.

Note that labels are selected randomly. Thus, the expectation occurring in this theorem is w.r.t this randomization.

Proof. In the setting of selective sampling, a model is updated whenever $Q_t Z_t = 1$. Given that $K > 2$ in multi-class setting, we bound as in Eq. (8),

$$\begin{aligned} &\sum_t Q_t Z_t (\|\mathbf{f}_t\|_2^2 - 2\mathbf{y}_t \cdot \mathbf{f}_t - \frac{Kr_t}{1+r_t} + 2h) \\ \leq &2h \sum_t Q_t Z_t \mathcal{L}(\mathbf{y}_t^\top U^\top \mathbf{m}_t) + h^2 tr(U^\top A_{\mathcal{U}_T} U). \end{aligned} \qquad (11)$$

If an error occurs ($y_t \neq \hat{y}_t$), $P_{W_t}^{\mathbf{m}_t}(y_t) \leq P_{W_t}^{\mathbf{m}_t}(y_t^{''})$. Thus,

$$-\mathbf{y}_t \cdot \mathbf{f}_t \overset{(4)}{=} P_{W_t}^{\mathbf{m}_t}(\hat{y}_t) - P_{W_t}^{\mathbf{m}_t}(y_t) \geq P_{W_t}^{\mathbf{m}_t}(\hat{y}_t) - P_{W_t}^{\mathbf{m}_t}(y_t^{''}).$$

Since $\|\mathbf{f}_t\|_2^2 \geq \frac{1}{2}\Delta_t^2$ and $P_{W_t}^{\mathbf{m}_t}(\hat{y}_t) - P_{W_t}^{\mathbf{m}_t}(y_t^{''}) = \Delta_t$, we bound,

$$\begin{aligned} &\sum_t Q_t Z_t (\|\mathbf{f}_t\|_2^2 - 2\mathbf{y}_t \cdot \mathbf{f}_t - \frac{Kr_t}{1+r_t} + 2h) \\ \geq &\sum_t Q_t Z_t (\frac{1}{2}\Delta_t^2 + 2\Delta_t - \frac{Kr_t}{1+r_t} + 2h) \\ \overset{(10)}{=} &\sum_t Q_t Z_t (\Theta_t + 2h). \end{aligned}$$

When an error occurs at trial $t \in \mathcal{M}$, the Θ_t can be positive ($\mathcal{M} \cap \mathcal{S}$) or negative ($\mathcal{M} \cap \mathcal{D}$). In the former case, $\mathbb{E}[Q_t] = \frac{2h}{2h+\Theta_t}$ is a random variable and we bound,

$$\mathbb{E}[Q_t Z_t (\Theta_t + 2h)] = \mathbb{E}[Z_t]\mathbb{E}[Q_t(\Theta_t + 2h)] = 2h\mathbb{E}[Z_t];$$

In the later case, $\mathbb{E}[Q_t] = 1$. Given $\frac{1}{2}\Delta_t^2 \geq 0$ and $\Delta_t \geq 0$, we have,

$$\begin{aligned} &\mathbb{E}[\sum_t Q_t Z_t (\frac{1}{2}\Delta_t^2 + 2\Delta_t - \frac{Kr_t}{1+r_t} + 2h)] \\ \geq &2h\mathbb{E}[Z_t] - \mathbb{E}[\sum_{t\in\mathcal{M}\cap\mathcal{D}} \frac{Kr_t}{1+r_t}] \end{aligned}$$

In summary,

$$\begin{aligned} &\sum_t Q_t Z_t (\Theta_t + 2h) \\ \geq &2h(\sum_{t\in\mathcal{M}\cap\mathcal{S}} \mathbb{E}[Z_t] + \sum_{t\in\mathcal{M}\cap\mathcal{D}} \mathbb{E}[Z_t]) - \mathbb{E}[\sum_{t\in\mathcal{M}\cap\mathcal{D}} \frac{Kr_t}{1+r_t}] \end{aligned}$$

With $\sum_{t\in\mathcal{M}} Z_t = |\mathcal{M}|$ and upper bound (11), we complete our proof. □

Remark 4. *The mistake bound of the CMOG on randomly selected labels is comparable with the bound of CMOG that learns all the labels. Similar to Theorem 1, the CMOG run on selected labels is bounded by the cumulative hinge loss suffered by U, $\log(T)$ (upper bound of $\sum_t \frac{r_t}{1+r_t}$) and $K\max_j \lambda_j$ (upper bound of $tr(U^\top A_{\mathcal{U}_T} U)$). Note that we use $\frac{Kr_t}{1+r_t}$ to let data "uncertainty" regularized by the class number. In addition, the CMOG run on selected labels can achieve a comparable mistake bound with the online algorithm OLLGC (i.e. Corollary 5, [14]) in binary classification, since $tr(U^\top A_{\mathcal{U}_T} U) \le \sum_{t\in\mathcal{M}} \|U^\top \mathbf{m}_t\|_2^2 \le \alpha tr(U^\top U) = \alpha tr(F^\top L F)$, where $\alpha = |\mathcal{M}|R^2$. Noted that this bound is incomparable with that in [6] since the Θ_t is different in two methods. In summary, the theoretical results present that an online algorithm learning on these selected labels could perform no worse than its fully-supervised counterpart. Thus above results theoretically demonstrate the efficacy of the proposed query method.*

C. Aggressive Learning

The CMOG is conservative, i.e., it will only update the model when an error occurs. To take advantage of the correctly predicted instances, we propose an aggressive version of selective sampling. We call our algorithm MSG, the Multi-class Selective Sampling on Graph, present in Algorithm 2. After observing a vertex $\mathbf{m}_t$ at round t, the MSG predicts its label with $W_t = A_t^{-1} B_{t-1}$ and then queries true label $\mathbf{y}_t$ with a probability of $\frac{2h}{2h+\max(0,\Theta_t)}$. It yields to stochastic query and deterministic query. When stochastic query (i.e. $\frac{2h}{2h+\max(0,\Theta_t)} < 1$) is issued, it is conservative to update model when an error occurs ($\hat{y}_t \neq y_t$). While a deterministic query is issued (i.e. $\frac{2h}{2h+\max(0,\Theta_t)} = 1$), we adopt an aggressive learning strategy, that is, we update even if no error occurs. Note that our model is different from [8], [1], since we perform a randomized query based on the predicted results of multiple classes.

The theoretical results below show the superiority of the aggressive algorithm compared to its conservative and fully-supervised counterpart CMOG (i.e. Algorithm 1). Besides the stochastic query trials $\mathcal{S}$ and deterministic query trials $\mathcal{D}$, we denote by $\mathcal{V}$ the set of trials for which there is an aggressive update but not a mistake (i.e., $y_t = \hat{y}_t$ and $\Theta_t < 0$) and let $V = |\mathcal{V}|$.

Theorem 5. *The algorithm MSG (Algorithm 2) runs on an arbitrary sequential nodes, then given $h > 0$, the following inequality holds for any $U \in \mathbb{R}^{n\times K}$,*

$$\mathbb{E}[M] \le \mathbb{E}[\sum_t Q_t Z_t \mathcal{L}(\mathbf{y}_t^\top U^\top \mathbf{m}_t)] + \frac{h}{2} tr(U^\top \mathbb{E}[A_{\mathcal{U}_T}] U) + \frac{1}{h}\mathbb{E}[\sum_{t\in D} \frac{Kr_t}{1+r_t}] - \mathbb{E}[V].$$

Algorithm 2 MSG: Multiclass Selective Sampling on Graph

Input: sequences of instance-label pair $(\mathbf{m}_t, \mathbf{y}_t), t = 1,\dots,T$, the parameters $\gamma > 0$ and $h > 0$.
Output: W_T
Initialize: $W_0 = 0$, $A_0 = \gamma I$ and $B_0 = \mathbf{0}$.
for $t = 1,\dots,T$ **do**
  Receive an input $\mathbf{m}_t$;
  Compute $A_t^{-1} = (A_{t-1} + \mathbf{m}_t\mathbf{m}_t^\top)^{-1}$;
  $\mathbf{f}_t = B_{t-1}^\top A_t^{-1}\mathbf{m}_t$;
  Predict $\hat{y}_t = \arg\max_{j\in[K]} P_{W_t}^{\mathbf{m}_t}(j)$;
  if $\Theta_t < 0$ (Definition 2) **then**
    Set $Q_t = Z_t = 1$ (i.e. deterministic query) and Query actual label $\mathbf{y}_t$;
  else
    Draw a Bernoulli random variable $Q_t \in \{0,1\} \sim \frac{2h}{2h+\max(0,\Theta_t)}$;
    if $Q_t = 1$ **then**
      Query actual label $\mathbf{y}_t$;
      Set $Z_t = 1$ if $\hat{y}_t \neq y_t$ ($Z_t = 0$, otherwise);
    end if
  end if
  $A_t = A_{t-1} + Q_t Z_t \mathbf{m}_t\mathbf{m}_t^\top$,
  $B_t = B_{t-1} + Q_t Z_t \mathbf{m}_t\mathbf{y}_t^\top$;
end for
$W_T = A_T^{-1} B_T$;

In addition, the expected number of queries is upper bounded by $\mathbb{E}[|\mathcal{D}| + \sum_{t\in\mathcal{S}} \frac{2h}{2h+\Theta_t}]$.

Proof. The update trials in algorithm 2 could be categorized into three groups,

$$\sum_t Z_t = |\mathcal{S}\cap\mathcal{M}| + |\mathcal{D}\cap\mathcal{M}| + |\mathcal{D}\cap\mathcal{V}|.$$

In the first case where an error occurs in randomized query with $\mathbb{E}[Q_t] = \frac{2h}{2h+\Theta_t}$ (i.e. $\mathcal{S}\cap\mathcal{M}$). Similar as Theorem 3,

$$\mathbb{E}[Q_t Z_t(\Theta_t + 2h)] = \mathbb{E}[Z_t]\mathbb{E}[Q_t(\Theta_t + 2h)] = \mathbb{E}[Z_t];$$

If an error incurs in a deterministic query (i.e. $t \in \mathcal{D}\cap\mathcal{M}$) with $\mathbb{E}[Q_t] = 1$, we bound,

$$\mathbb{E}[Q_t Z_t(\frac{1}{2}\Delta_t^2 + 2\Delta_t - \frac{Kr_t}{1+r_t} + 2h)] \ge 2h\mathbb{E}[Z_t] - \frac{Kr_t}{1+r_t};$$

Now we consider the third case where the updates were performed with no mistake, i.e., $\Theta_t \le 0$, and by definition,

$$\Theta_t \le 0 \Rightarrow 0 \le \Delta_t \le 2\sqrt{1 + \frac{Kr_t}{2(1+r_t)}} - 2. \qquad (12)$$

If no mistake incurs ($y_t = \hat{y}_t$), we have $\mathbf{y}_t\cdot\mathbf{f}_t = P_{W_t}^{\mathbf{m}_t}(\hat{y}_t) - \max_{j\neq\hat{y}_t} P_{W_t}^{\mathbf{m}_t}(j) = P_{W_t}^{\mathbf{m}_t}(\hat{y}_t) - P_{W_t}^{\mathbf{m}_t}(y_t'') = \Delta_t$. Thus,

$$\begin{aligned}&\mathbb{E}[Q_t Z_t(\frac{1}{2}\Delta_t^2 - 2\Delta_t - \frac{Kr_t}{1+r_t} + 2h)]\\ =&\mathbb{E}[Z_t(\frac{1}{2}\Delta_t^2 - 2\Delta_t + \frac{Kr_t}{1+r_t} - \frac{2Kr_t}{1+r_t} + 2h)]\end{aligned}$$

Let $C(\Delta_t, r_t) = \frac{1}{2}\Delta_t^2 - 2\Delta_t + \frac{Kr_t}{1+r_t}$. Whenever $\frac{Kr_t}{1+r_t} \geq 2$, $C(\Delta_t, r_t) \geq 0 (\forall \Delta_t \geq 0)$. If $\frac{Kr_t}{1+r_t} < 2$, let $C(\Delta_t, r_t)$ be a quadratic equation with two non-negative roots and a minima, $2 - 2\sqrt{1 - \frac{Kr_t}{2(1+r_t)}}$. We observe this smaller root is higher than the upper bound in (12), that makes $C(\Delta_t, r_t) \geq 0$ in the trials $\mathcal{D} \cap \mathcal{V}$. Thus, we bound,

$$\mathbb{E}[Q_t Z_t (\frac{1}{2}\Delta_t^2 - 2\Delta_t - \frac{Kr_t}{1+r_t} + 2h)] \geq 2h\mathbb{E}[Z_t] - \frac{2Kr_t}{1+r_t}.$$

To summarize,

$$\mathbb{E}[\sum_t Q_t Z_t (\Delta_t^2 - 2\mathbf{f}_t \cdot \mathbf{y}_t - \frac{Kr_t}{1+r_t} + 2h)]$$
$$\geq 2h \sum_{t \in \mathcal{M}} \mathbb{E}[Z_t] + 2h \sum_{t \in \mathcal{D} \cap \mathcal{V}} \mathbb{E}[Z_t] - \sum_{t \in \mathcal{D}} \mathbb{E}[\frac{2Kr_t}{1+r_t}].$$

Equipped with upper bound as Eq. (11), we complete the proof. □

Remark 6. *The upper bound of the aggressive algorithm MSG is expected to be lower than CMOG that learns on all the labels (Theorem 1) and CMOG on the selected labels (Theorem 3), due to the deduction of $\mathbb{E}[\mathcal{V}]$ from the bound. In summary, the theoretical analysis demonstrate that the MSG, in expectation, can achieve a better performance than its conservative and fully-supervised counterparts, which can be regarded as a theoretical support for the aggressive method.*

Discussion: To further understand the aggressive algorithm, we analyze under what condition an aggressive query will be conducted. An aggressive query is issued when $\Theta_t \leq 0$ (i.e.,$\Theta_t \leq 0 \Rightarrow \Delta_t \leq \theta(K, r_t) = 2\sqrt{1 + \frac{Kr_t}{2(1+r_t)}} - 2$). If the margin Δ_t is less than $\theta(K, r_t)$, a deterministic query is issued, while Δ_t is above $\theta(K, r_t)$, a label is queried randomly with a probability less than 1. We observe that the upper bound of $\theta(K, r_t)$ increases with r_t. When $r_t = 0$, that is, current instance is observed before, the label would be queried deterministically in case its margin is 0 ($\Delta_t \leq \theta(K, r_t = 0) = 0$, i.e., an extreme case that the current model is unable to predict its label). However, if $r_t = 1$ (i.e., little knowledge to current input), the learner would query aggressively whenever its margin does not exceed $\theta(K, r_t = 1) = \sqrt{4 + K} - 2$, a threshold far from the boundary.

V. EXPERIMENTAL RESULTS

In this section, we first introduce experimental dataset and evaluation metrics. Then we present the empirical results to validate the proposed algorithms. Our experiments are designed to answer two questions: (i) if the proposed randomized query is effective to reducing the amount of labeled data significantly while maintain comparable performance? (ii) if the aggressive strategy achieves a better predictive performance at the cost of more queried number?

A. Data Sets and Evaluation Metrics

Data Sets: Four real-world graph data sets are used in the experiment to evaluate the approaches.
(a) Coauthor[2] extracted from *DBLP* database is an undirected co-author graph in which 1711 authors are denoted as vertices while their co-authored relationship are treated as the edges. The authors are classified in four classes in terms of research topic: "data mining", "machine learning", "information retrieval" and "databases". **(b)** Cora[3] is a citation network including 2485 scientific publications and 5429 citation links. The publications as vertices are related to seven domains: "Case based", "Genetic Algorithms", "Rule Learning", "Probabilistic Methods", "Neural Networks", "Reinforcement Learning", et al. **(c)** IMDB[4] is a movie organization that presses up-to-date movie information. The IMDB links total 17046 movies with their co-actor associations. The movies as vertices in graph are categorized into four genres: "Action", "Romance", "Animation" and "Thriller". **(d)** PubMed[5] is also a citation graph related to diabetes research. The PubMed collects 44338 publication citations among 19717 scientific publications and labels the publications with one of three types of diabetes.
The graph data is supposed to be undirected and connected. If the edges are directed, we transform them into undirected graphs via $\mathbf{S} \leftarrow \max(\mathbf{S}, \mathbf{S}^\top)$. If the graphs are disconnected, the biggest connected subgraph is chosen for study.

Evaluation Measures: We evaluate the performance of baselines and our algorithms with two measurements:
i) *cumulative error rate*, reflecting the prediction accuracy of online algorithm; **ii)** *number of queried labels*, reflecting the label efficiency of query method. Note that a small value of above measures indicates a better performance of a method. In order to compare these algorithms fairly, we randomly shuffle the ordering of samples for each dataset. We repeat each experiment 20 times and calculate the average results.

Baselines and Parameter Setting: We compare the proposed algorithms with state-of-the-art baselines. The algorithms we study and their parameter settings are summarized as follows. **(1)** GPA: a first order nonparametric online learning algorithm on graph [15]. Note that the perceptron algorithm is not affected by the step-size. **(2)** BBQ/BBQ$_\epsilon$: The two algorithms are the BBQ query criterion [5] and its modification version [1]. The intuition behind this rule is that at the rounds where the label is not queried, it guaranteed that the regret bound is at most ϵ. The parameter ϵ is tuned with grid $\{10^{-5}, \ldots, 10\}$ in our experiment. **(3)** DGS: This query

[2]https://snap.stanford.edu/data/com-DBLP.html
[3]http://www.cs.umd.edu/ sen/lbc-proj/data/
[4]http://www.imdb.com/
[5]http://www.cs.umd.edu/projects/linqs/projects/lbc/

TABLE I
COMPARISON OF THE MULTI-CLASS ALGORITHMS. GPA AND CMOG ARE ONLINE ALGORITHMS.

Algorithm	*Coauthor*		*Cora*	
	Error rate	# Queried nodes	Error rate	# Queried nodes
GPA	0.5474±2.66e-4	1711	0.5849±2.03e-4	2485
BBQ	0.3013±3.55e-5	1371±168.7	0.1929±1.73e-5	1635.1±294.0
BBQ_ϵ	0.3028±3.03e-5	1711	0.1936±1.95e-5	2485
DGS	0.3096±3.29e-5	1711	0.1941±2.35e-5	2485
CMOG	0.3096±3.29e-5	1711	0.1940±2.33e-5	2485
MSG	**0.2956±6.63e-5**	**870.7±251.9**	**0.1926±2.34e-5**	**884.95±289.15**
Algorithm	*IMDB*		*PubMed*	
	Error rate	# Queried nodes	Error rate	# Queried nodes
GPA	0.6870±4.6e-5	17046	0.5795±2.9e-6	19717
BBQ	0.5468±1.69e-6	10033±467.2	0.2217±1.9e-6	10352±2536.4
BBQ_ϵ	0.5068±9.9e-6	16983±801.3	0.2217±3.1e-6	8986.3±838.3
DGS	0.5066±9.1e-6	17046	0.2265±1.46e-6	19717
CMOG	0.5576±6.88e-5	17046	0.2265±1.5e-6	19717
MSG	**0.5043±7.46e-5**	**3750.3±255.1**	**0.2158±1.07e-5**	**936.29±210.07**

criterion is a nonparametric rule for binary classification [11] and adapts into multi-class setting in [1]. It takes both previous covariances and the observed labels into account. The intuition behind this rule is that on rounds where the label y_t is not queried, it guaranteed that either $\hat{y}_t = y_t^*$, or the regret is small. **(4)** CMOG and MSG: two second-order algorithm in multi-class setting. CMOG is a conservative online algorithm while MSG is an aggressive algorithm that queries label with a randomized method. We set $\gamma = 1$ to avoid overfiiting and tune the parameter h with grid $\{10^{-4}, \ldots, 1\}$ on a held-out random shuffle.

B. Comparison Evaluation

The experimental results are present in Table I. We found that MSG outperforms all baselines consistently across all data sets. We also show the results in terms of learning epoches in Figure 1. In all subfigures, the cumulative error rate and queried number along the learning epochs are both averaging over 20 times of shuffling order.

First of all, we observe that the improvement of the CMOG over GPA are always significant on all data sets. This is consistent with previous observations in online learning: second-order algorithms are generally better than first-order algorithm . The reason is due to the covariance matrix A_t which has a spectral structure to correlate with a best estimator for observed instances [4]. MSG always enjoys smaller or comparable error rates than BBQ_ϵ and DGS with much fewer queried number. The good performance generally is due to two reasons. First, the proposed randomized query approach improves the efficiency of the labeling. Second, thanks to the aggressive learning, the MSG achieves a convergence stage quickly with informative labels, thus the query rate is reduced further when learner has sufficient knowledge of data. The results in figure 1 indicate the MSG queries a small number of labels while maintains the quality of classification model.

C. Evaluation on Varied Ratios of Queries

We study the impact of h with respect to query ratio of the MSG. Basically, the smaller *h* is, the fewer the number of queries is. Specifically, we set *h* to $\{10^{-4}, 10^{-3}, \ldots, 1\}$, and run MSG for 20 times under each h. We calculate the average ratio of queried nodes under different values of *h*. The comparison results in Figure 2 show that MSG achieves better or comparable performance consistently under different ratios of queried labels. This validates the label-efficiency of our proposed confidence score Θ_t that can adaptively prioritize informative labels to optimize the model. We also observe that the MSG outperforms BBQ significantly over all query ratios. The reason is that MSG considering "Δ_t" will query the vertices close to current boundary, while these labels are omitted in BBQ. The better results in figure 2 demonstrate that these "small-margin" instances are useful to optimize the multi-class classifier.

D. Evaluation on Low-rank Approximation

The low-rank vertex representation $\mathbf{m} \in \mathbb{R}^d$ is used to build a scalable online model in our experiments. To study the impact of low-rank approximation on the proposed algorithms, we tune the rank d in the grid {10, 100, 250, 500, 750, 1000}. We use Coauthor and Cora as a case study since similar observations are obtained on other data sets. The results in Figure 3 present that MSG achieves a better or comparable performance than other baselines consistently under different rank approximation. Obviously with a higher rank, the performance becomes better in terms of error rate. However, to achieve a better prediction accuracy, algorithms need a high number of queries and high-rank inputs, which demands a more labeling and computational cost. It motivates us to select a proper rank d to achieve a balance. Therefore, we chose $d = 100$ in the rest of experiments, since in this setting

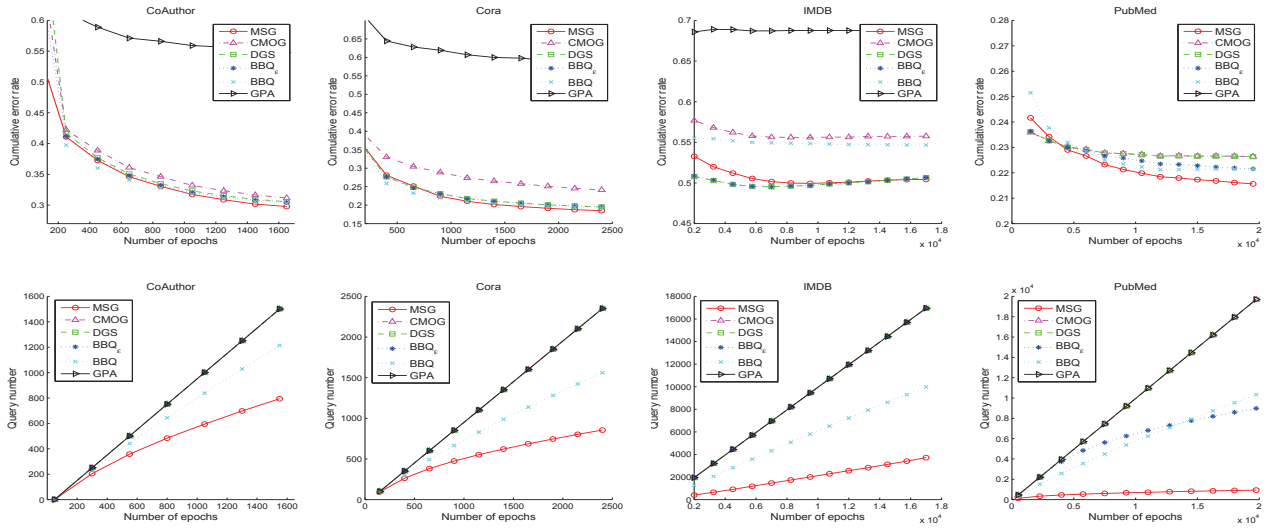

Fig. 1. Cumulative error rate and Query number with respective to online learning rounds on four datasets.

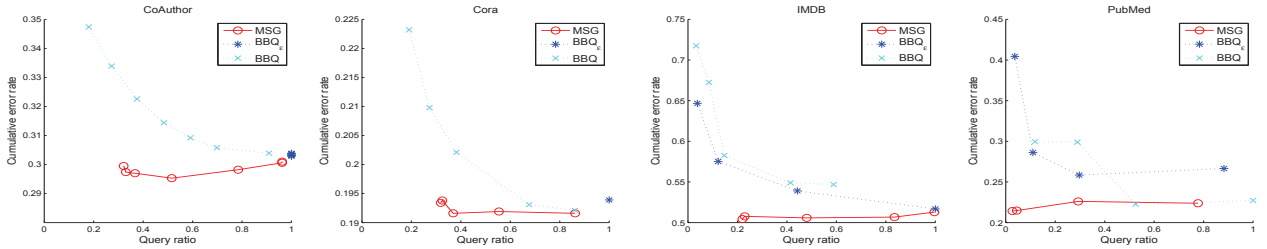

Fig. 2. A comparison among BBQ, BBQ$_\epsilon$ and MSG with respect to different ratios of queried nodes.

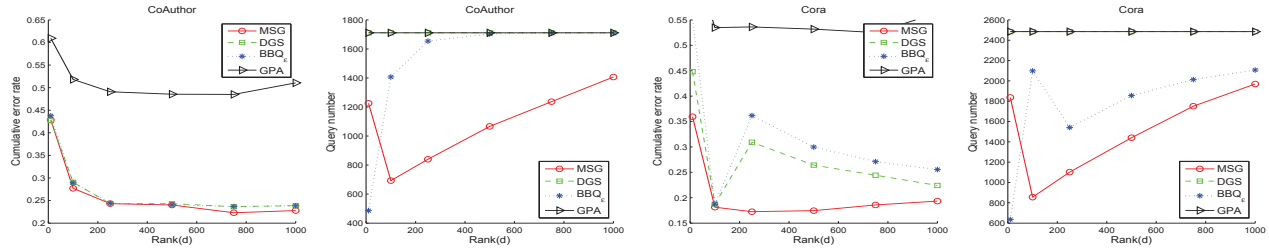

Fig. 3. A case study of low rank impact on performance.

the algorithms perform well while number of queries is small.

VI. CONCLUSIONS

In this paper, we proposed a new framework for multi-class online learning, leading to a scalable algorithm to tackle multi-class classification on graphs. To save the labeling cost, we presented a novel randomized query technique to prioritize the labels. Besides, we introduced an aggressive selective sampling algorithm to take full advantage of these wasted labels in existing conservative algorithms. The theoretical results demonstrated the efficacy of the proposed algorithms in terms of the expected mistake bound and query ratio.

The encouraging empirical results on several real-world datasets also indicated that 1) the MSG is able to achieve comparable or better predictive performance by querying a significantly small amount of labeled data; and that 2) the aggressive selective sampling scheme can further reduce the query rate, achieving a convergence stage rapidly.

References

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

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

[3] M. S. Bartlett. An inverse matrix adjustment arising in discriminant analysis. *The Annals of Mathematical Statistics*, pages 107–111, 1951.

[4] N. Cesa-Bianchi, A. Conconi, and C. Gentile. A second-order perceptron algorithm. *SIAM Journal on Computing*, 34(3):640–668, 2005.

[5] N. Cesa-Bianchi, C. Gentile, and F. Orabona. Robust bounds for classification via selective sampling. In *ICML*, pages 121–128, 2009.

[6] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Worst-case analysis of selective sampling for linear classification. *JMLR*, 7:1205–1230, 2006.

[7] O. Chapelle, B. Schölkopf, A. Zien, et al. *Semi-supervised learning*, volume 2. MIT press Cambridge, 2006.

[8] K. Crammer. Doubly aggressive selective sampling algorithms for classification. In *AIStat*, pages 140–148, 2014.

[9] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *JMLR*, 7:551–585, 2006.

[10] K. Crammer, A. Kulesza, and M. Dredze. Adaptive regularization of weight vectors. In *NIPS*, pages 414–422, 2009.

[11] O. Dekel, C. Gentile, and K. Sridharan. Selective sampling and active learning from single and multiple teachers. *JMLR*, 13(1):2655–2697, 2012.

[12] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of artificial intelligence research*, pages 263–286, 1995.

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

[14] Q. Gu, C. Aggarwal, J. Liu, and J. Han. Selective sampling on graphs for classification. In *ACM SIGKDD*, pages 131–139, 2013.

[15] M. Herbster and M. Pontil. Prediction on a graph with a perceptron. In *NIPS*, pages 577–584, 2006.

[16] M. Herbster, M. Pontil, and L. Wainer. Online learning over graphs. In *ICML*, pages 305–312, 2005.

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

[18] P. Yang, G. Li, P. Zhao, X. Li, and S. D. Gollapalli. Learning correlative and personalized structure for online multi-task classification. In *SDM*, 2016.

[19] P. Yang, X. Li, M. Wu, C.-K. Kwoh, and S.-K. Ng. Inferring gene-phenotype associations via global protein complex network propagation. *PloS one*, 6(7):e21502, 2011.

[20] P. Yang, X.-L. Li, J.-P. Mei, C.-K. Kwoh, and S.-K. Ng. Positive-unlabeled learning for disease gene identification. *Bioinformatics*, 28(20):2640–2647, 2012.

[21] P. Yang and P. Zhao. A min-max optimization framework for online graph classification. In *CIKM*, pages 643–652. ACM, 2015.

[22] P. Yang, P. Zhao, V. W. Zheng, and X. Li. An aggressive graph-based selective sampling algorithm for classification. In *ICDM 2015*, pages 509–518, 2015.