An Aggressive Graph-based Selective Sampling Algorithm for Classification

Peng Yang*, Peilin Zhao*, Vincent W. Zheng† and Xiao-Li Li*
* Institute for Infocomm Research, A*STAR, 138632, Singapore
† Advanced Digital Sciences Center, UIUC, 138632, Singapore
Email: {yangp, zhaop, xlli}@i2r.a-star.edu.sg, vincent.zheng@adsc.com.sg

***Abstract*—Traditional online learning algorithms are designed for *vector* data only, which assume that the labels of all the training examples are provided. In this paper, we study *graph* classification where only *limited* nodes are chosen for labelling by selective sampling. Particularly, we first adapt a spectral-based graph regularization technique to derive a novel online learning linear algorithm which can handle graph data, although it still queries the labels of all nodes and thus is not preferred, as labelling is typically time-consuming. To address this issue, we then propose a new confidence-based query method for selective sampling. The theoretical result shows that our online learning algorithm with a fraction of queried labels can achieve a mistake bound comparable with the one learning on all labels of the nodes. In addition, the algorithm based on our proposed query strategy can achieve a mistake bound better than the one based on other query methods. However, our algorithm is *conservative* to update the model whenever error happens, which obviously wastes training labels that are valuable for the model. To take advantage of these labels, we further propose a novel *aggressive* algorithm, which can update the model aggressively even if no error occurs. The theoretical analysis shows that our aggressive approach can achieve a mistake bound better than its conservative and fully-supervised counterpart, with substantially fewer queried times. We empirically evaluate our algorithm on several real-world graph datasets and the experimental results demonstrate that our method is highly effective.**

***Keywords*—Online Learning, Selective Sampling, Graph Node Classification**

I. Introduction

Graphs, including social network (e.g. *Facebook*[1]), communication network and citation network, have attracted significant attention due to their wide range of applications. Specifically, there are increasing needs to classify the graph nodes into different classes, e.g. potential customers (positive class) or non-customers (negative class) in a social network. To solve these problems, the classification models can be learned using a set of training examples (i.e. some node-label pairs) from the graphs, which have been studied in both *offline* [8] and *online* settings [17]. *Offline* learning algorithms can query the labels of all stored nodes whenever they need, thus demanding a high memory and storage space. *Online* learning, on the other hand, can query the labels in a sequential order and only accesses the labels of the current nodes. Additionally, after updating the models, the current inputs are discarded and will not be stored for further query. As a result, online learning algorithms are very efficient and scalable, which are particularly useful to handle big graph data [17].

[1]http://www.facebook.com

Nevertheless, online learning typically assumes that the labels of all the nodes are provided. This clearly limits its application in many real-world scenarios, as it is costly to label all the nodes. As such, some *active* online learning algorithms, namely *selective sampling* [12], [15], [7], [6], [21], [17], [14], have been proposed recently. The objective of selective sampling is to achieve a good trade-off between the classification error and query times. Specifically, a selective sampling will decide not only whether to query the label of the current input, but also how to update a model after the true label is revealed. One algorithm is called *conservative* if it only updates a model when a prediction error happens, while an *aggressive* algorithm can still update with the labels even if no error occurs. Note currently all existing selective samplings on graphs are conservative [14], thus they cannot leverage the correctly predicted labels to improve a prediction model.

Based on above observations, we study a better selective sampling algorithm on graphs. We first adapt graph kernel [23] into a Laplacian Regularized Least Square (LapRLS) [3] to derive an online linear algorithm on graph data (COLA). The COLA can leverage the update scale, although it has to query every node during its learning process, which is not efficient in big graph applications. To make it scalable on large-scale datasets, we propose a conservative graph-based *selective sampling* algorithm (CGS), which queries only a fraction of the node labels. The algorithm decides to query a label based on a novel *confidence* method, which considers not only the absolute classification *margin*, but also the prediction *uncertainty*. The theoretical analysis demonstrates that the CGS learning on a fraction of queried labels can achieve, as expected, a comparable mistake bound with the one of CGS learning on all the labels. In addition, we prove that the CGS based on our proposed query method can achieve a mistake bound better than the one based on other query strategies [7], [9], which can be treated as a theoretical support for our proposed query approach. Nevertheless, the CGS is *conservative* to update the model when error happens, thus wasting the correctly predicted labels that are still useful for the model. To take advantage of these labels, we propose an *aggressive* version of graph-based selective sampling algorithm (AGS). The AGS hybrids the conservative update with the aggressive one, which can update aggressively even if no error occurs. Theoretical analysis shows that the AGS can achieve a mistake bound better than not only the conservative algorithm CGS but also the fully-supervised online algorithm COLA. Furthermore, extensive experiments on multiple graph data demonstrate our proposed algorithm is better than the state-of-the-art techniques with substantial fewer queries.

The remaining parts of this paper are organized as follows. First, Section 2 provides the related work and Section 3 introduces the problem settings. Then, the proposed algorithm and theoretical analysis are presented in section 4. Next, Section 5 empirically evaluates various algorithms. Finally, Section 6 concludes the paper, followed by the Appendix.

II. Related Work

In this section, we briefly introduce selective sampling in the setting of vector data and graph data, respectively.

Vector-based Setting: Selective sampling on vector-based data is a combination of online learning [28], [27], [18] and active learning [22], [13], [26], [20]. These algorithms generally can be categorized into first-order algorithms [10], [4], [7] and second-order algorithms [21], [6]. First-order algorithms make a query decision based on a concept of "margin" [10]. Specifically, these methods query a label when the input lies near the current hypothesis (a small margin). However, they are unable to optimize the direction and scale of model update. To address this issue, we have witnessed several second-order algorithms [21], [6] in recent years, which track a spectral structure of observed inputs and decide to query based on a notation of "uncertainty". They usually query a label if the current model has little knowledge in the direction of the input.

While many studies w.r.t. label selection have been done in both first-order and second-order setting, few methods [7], [21], [9] combine these two quantities to make a query decision. Unlike the deterministic query in [21], we propose a randomized query strategy, i.e. *confidence-based query*, leveraging these two quantities together. We prove that selective sampling based our query method can achieve a better performance than [7], [9].

Graph-based Setting: The setting of selective sampling on graph is graph Laplacian for *transductive* learning, where the whole graph structure is provided, and the learner observes graph nodes in a streaming order. Unlike the vector-based setting with i.i.d. assumption on the data, few algorithms are designed for graphs, among which the pioneering work is GPA, a first-order fully-supervised online algorithm [16]. Besides, OLLGC and SSLGC [14], two second-order algorithms, are the most relevant to our work. They reveal a true label only when the "uncertainty" of input is above a threshold. Both two algorithms are conservative to update model, thus they waste the correct predicted labels. Several recent works focus on the active-transductive learning [19], [25]. However, these methods are offline settings where labels are queried from a pool of observed inputs, while online setting is allowed to query current inputs, after that, the observed inputs are discarded and cannot be queried.

Despite extensive works have been done in both the fields of selective sampling under vector-based and graph-based setting, to the best of our knowledge, we are the first to propose a graph-based selective sampling that hybrids the conservative update with aggressive one for classification. Specifically, we first derive a new online linear model based on a spectral graph regularization technique. Then we propose a confidence-based query that can effectively make full use of informative labels in a conservative and aggressive way.

III. Problem Setting

In this section, we first introduce our notations. Then we review the Laplacian Regularized Least Squares (LapRLS) [3] to derive a linear model (i.e. $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$) for graph classification.

A. Notations

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 bold-face 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 $\mathbf{A}^\dagger$. In addition, a diagonal matrix is denoted as $\mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ with diagonal elements $\sigma_i, i \in [1, n]$. Finally, the ℓ_2-norm of vector $\mathbf{w} \in \mathbb{R}^d$ is defined as $\|\mathbf{w}\| = \sqrt{\sum_{i=1}^d w_i^2}$.

B. Laplacian Regularized Least Squares

A graph is defined as $G = (V,E)$ 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 a $n \times n$ adjacency matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$, where real-valued element S_{ij} is measured by the affinity of a pair (v_i, v_j). In this paper, we assume graph G is connected and undirected. Given that $\mathbf{D}$ is the diagonal matrix with diagonal elements $D_{ii} = \sum_k S_{ik}$, we define $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is graph Laplacian, $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ ($\mathbf{v}_i \in \mathbb{R}^n$) are its eigenvectors, and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ ($0 = \lambda_1 \leq \ldots \leq \lambda_n \leq 2$) are its eigenvalues.

Graph regularization [23] assumes a label smooth over the graph, which formats as $\frac{1}{2}\sum_{i,j=1}^n S_{ij}(f_i - f_j)^2 = \mathbf{f}^\top \mathbf{L} \mathbf{f}$, where $\mathbf{f} = [f_1, \ldots, f_n]^\top \in \mathbb{R}^n$, $f_i : v_i \to \mathbb{R}$ is a predicted value on the node v_i. Intuitively, the regularization suffers from a loss if neighboring nodes are mapped far part. In the setting of binary classification, LapRLS solves the following problem based on graph regularization,

$$\min_{\mathbf{f}} \frac{1}{2}\|\mathbf{f} - \mathbf{y}\|^2 + \frac{\mu}{2}\mathbf{f}^\top \mathbf{L} \mathbf{f} + \frac{\gamma}{2}\|\mathbf{f}\|^2 \tag{1}$$

where $\mathbf{y} = [y_1, y_2, \ldots, y_n]^\top$ is a true label vector, $\mu > 0$ is a regularized parameter for graph Laplacian and $\gamma > 0$ for result generalization. In the setting of graph classification, the real-valued function should satisfy: 1). the values of function $\mathbf{f}$ for labeled nodes should be close to the given labels for that nodes; 2). nodes should satisfy label smoothness on the whole graph, that is, the points nearby in the graph should have similar labels; 3). function values on all nodes should be generalized to solve ill-posed problem without overfitting.

Motivated by graph kernel $\mathbf{f} = \mathbf{L}^\dagger \beta$ [23] ($\beta \in \mathbb{R}^n$), we introduce a linear model for graph classification,

Definition 1. *Given a graph kernel $\mathbf{f} = \mathbf{L}^\dagger \beta$ where $\mathbf{L}^\dagger = \sum_{i=1}^n \frac{1}{\lambda_i} \mathbf{v}_i \mathbf{v}_i^\top$, the model can turn to a linear regression function,*

$$\mathbf{f} = \mathbf{X}^\top \mathbf{w}, \tag{2}$$

where $\mathbf{w} = [\frac{1}{\lambda_1}\mathbf{v}_1, \ldots, \frac{1}{\lambda_n}\mathbf{v}_n]^\top \beta$ and $\mathbf{X} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]^\top$.

We define $\mathbf{w} \in \mathbb{R}^n$ is a weight vector and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ is a data matrix where node $\mathbf{x}_j = [v_{1j}, \ldots, v_{nj}]^\top \in \mathbb{R}^n$, $j \leq n$ is one column of $\mathbf{X}$. Equipped with Eq. (2), the graph function in Eq. (1) can turn to an equivalent formulation,

$$\arg\min_{\mathbf{w}} \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i^\top \mathbf{w} - y_i)^2 + \frac{\mu}{2}\mathbf{w}^\top \mathbf{X}\mathbf{L}\mathbf{X}^\top \mathbf{w} + \frac{\gamma}{2}\sum_{i=1}^{n}(\mathbf{x}_i^\top \mathbf{w})^2. \quad (3)$$

Lemma 1. $\mathbf{L}$ *is a positive semidefinite matrix, where* $\{(\mathbf{v}_i, \lambda_i)\}(i \leq n)$ *is the eigensystem of* $\mathbf{L}$. *Assume* $\mathbf{X} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]^\top$, *then*

$$\begin{aligned}&\mathbf{X}\mathbf{L}\mathbf{X}^\top \\ =&[\mathbf{v}_1, \ldots, \mathbf{v}_n]^\top \mathbf{L}[\mathbf{v}_1, \ldots, \mathbf{v}_n] = \mathit{diag}(\lambda_1, \ldots, \lambda_n).\end{aligned}$$

As a result, we can turn a Laplacian term to a simple form,

$$\mathbf{w}^\top \mathbf{X}\mathbf{L}\mathbf{X}^\top \mathbf{w} = \mathbf{w}^\top \mathrm{diag}(\lambda_1, \ldots, \lambda_n)\mathbf{w} = \sum_{i=1}^{n} \lambda_i w_i^2.$$

Then the optimization problem in Eq. (3) can be rewritten,

$$\arg\min_{\mathbf{w}} \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i^\top \mathbf{w} - y_i)^2 + \frac{\mu}{2}\mathbf{w}^\top \mathbf{\Lambda}_n \mathbf{w} + \frac{\gamma}{2}\sum_{i=1}^{n}(\mathbf{x}_i^\top \mathbf{w})^2, \quad (4)$$

where we define $\mathbf{\Lambda}_n = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$. To make our algorithm scalable to large graphs, we propose a low-rank approximation for graph regularization, that is, we employ a low-dimensional data i.e., a low-rank input $\mathbf{X}_d = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ where $\mathbf{x}_i \in \mathbb{R}^d$ and graph Laplacian $\mathbf{\Lambda}_d \in \mathbb{R}^{d\times d}$ where $d \ll n$, and use the low-rank data representation in next section.

IV. Algorithms

In this section, we first propose an online learning algorithm on the graph. Then we present the objective of selective sampling, followed with a conservative algorithm and an aggressive algorithm. Finally, we introduce a low rank approximation analysis.

A. Online Learning on Graph

The purpose of online learning is to minimize the cumulative loss over sequential nodes. Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ $(T \leq n)$ be a sequence of nodes, where $\mathbf{x}_t \in \mathbb{R}^d$ is a column of a low-rank matrix $\mathbf{X}_d \in \mathbb{R}^{d\times n}(d \ll n)$ and $y_t \in \{\pm 1\}$ with $t < T$, an online version of graph regularization is defined:

$$G(\mathbf{w}) = \frac{1}{2}\sum_{t=1}^{T}(\mathbf{x}_t^\top \mathbf{w} - y_t)^2 + \frac{\mu}{2}\mathbf{w}^\top \mathbf{\Lambda}_d \mathbf{w} + \frac{\gamma}{2}\sum_{t=1}^{T}(\mathbf{x}_t^\top \mathbf{w})^2.$$

We next solve $G(\mathbf{w})$ over $\mathbf{w} \in \mathbb{R}^d$ in the following lemma, starting with the notation below,

$$\mathbf{A}_T = \mu\mathbf{\Lambda}_d + \sum_{t=1}^{T}(1+\gamma)\mathbf{x}_t\mathbf{x}_t^\top, \quad \mathbf{b}_T = \sum_{t=1}^{T} y_t \mathbf{x}_t. \quad (5)$$

Lemma 2. *For all* $T \geq 1$, $G(\mathbf{w}) = \frac{1}{2}\sum_{t=1}^{T}(\mathbf{x}_t^\top \mathbf{w} - y_t)^2 + \frac{\mu}{2}\mathbf{w}^\top \mathbf{\Lambda}_d \mathbf{w} + \frac{\gamma}{2}\sum_{t=1}^{T}(\mathbf{x}_t^\top \mathbf{w})^2$ *is minimal at an unique point* $\mathbf{w}_T \in \mathbb{R}^d$ *given by,*

$$\mathbf{w}_T = \mathbf{A}_T^{-1}\mathbf{b}_T, \quad G(\mathbf{w}_T) = \sum_{t=1}^{T} y_t^2 - \mathbf{b}_T^\top \mathbf{A}_T^{-1}\mathbf{b}_T. \quad (6)$$

Proof: From

Algorithm 1 COLA: A Conservative Online Linear Model on Graph for Classification

Input: Adjacency matrix $\mathbf{S}$, rank d, and regularization parameter μ and λ.
Output: $\mathbf{w}_T$
Compute $\mathbf{L} = \mathbf{D} - \mathbf{S}$ and $\mathbf{X}$ from $\mathbf{L}$
Initialize: $\mathbf{A}_0 = \mu\mathbf{\Lambda}_d$, $\mathbf{b}_0 = \mathbf{0}$, $\mathbf{w}_0 = \mathbf{0}$
for $t = 1, \ldots, T$ **do**
  Receive $\mathbf{x}_t \in R^d$
  Compute $\mathbf{A}_t^{-1} = (\mathbf{A}_{t-1} + (1+\lambda)\mathbf{x}_t\mathbf{x}_t^\top)^{-1}$
  Predict $\hat{y}_t = \mathrm{sign}\ (\mathbf{b}_{t-1}^\top \mathbf{A}_t^{-1}\mathbf{x}_t)$
  Query the actual label y_t
  if $\hat{y}_t \neq y_t$ **then**
    Update $\mathbf{A}_t = \mathbf{A}_{t-1} + (1+\lambda)\mathbf{x}_t\mathbf{x}_t^\top$
    Update $\mathbf{b}_t = \mathbf{b}_{t-1} + y_t\mathbf{x}_t$.
  else
    $\mathbf{A}_t = \mathbf{A}_{t-1}, \mathbf{b}_t = \mathbf{b}_{t-1}$
  end if
end for
$\mathbf{w}_T = \mathbf{A}_T^{-1}\mathbf{b}_T$

$$\begin{aligned}G(\mathbf{w}) &= \frac{1}{2}\sum_{t=1}^{T}(\mathbf{x}_t^\top \mathbf{w} - y_t)^2 + \frac{\mu}{2}\mathbf{w}^\top \mathbf{\Lambda}_d \mathbf{w} + \frac{\gamma}{2}\sum_{t=1}^{T}(\mathbf{x}_t^\top \mathbf{w})^2 \\ &= \frac{1}{2}\sum_{t=1}^{T}(y_t^2 - 2\mathbf{w}^\top(y_t\mathbf{x}_t)) + \frac{1}{2}\mathbf{w}^\top(\mu\mathbf{\Lambda}_d + \sum_{t=1}^{T}(1+\gamma)\mathbf{x}_t\mathbf{x}_t^\top)\mathbf{w} \\ &\overset{(5)}{=} \frac{1}{2}\sum_{t=1}^{T} y_t^2 - \mathbf{w}^\top \mathbf{b}_T + \frac{1}{2}\mathbf{w}^\top \mathbf{A}_T \mathbf{w},\end{aligned}$$

we have $\nabla_{\mathbf{w}} G(\mathbf{w}) = \mathbf{A}_T\mathbf{w} - \mathbf{b}_T = 0$, $\nabla^2_{\mathbf{w}} G(\mathbf{w}) = \mathbf{A}_T$. $G(\mathbf{w})$ is convex in $\mathbf{w}$ due to $\mathbf{A}_T \geq 0$, and it achieves a minimal point if $\nabla_{\mathbf{w}} G(\mathbf{w}) = 0$, that is, $\mathbf{w}_T = \mathbf{A}_T^{-1}\mathbf{b}_T$. Substituting the solution back into $G(\mathbf{w})$, and we have

$$G(\mathbf{w}_T) = G(\mathbf{A}_T^{-1}\mathbf{b}_T) = \sum_{t=1}^{T} y_t^2 - \mathbf{b}_T^\top \mathbf{A}_T^{-1}\mathbf{b}_T.$$

■

In Lemma 2, we obtain an optimal linear solution $\mathbf{w}_T$ for online learning on graph. Inspired by [1], we take advantage of current node to predict its label with $\hat{y}_t = \mathrm{sign}(\mathbf{b}_{t-1}^\top \mathbf{A}_t^{-1}\mathbf{x}_t)$. Moreover, it is inefficient for online algorithm to update in each iteration. To address this issue, we make use of a conservative strategy [5] to let model update when an error $(y_t \neq \hat{y}_t)$ occurs. We call the algorithm COLA, conservative online linear model on graph. Note that our update format is different from OLLGC [14] and SOP [5], since we derive an online linear model that is more flexible to leverage the update scale with μ and γ: When $\frac{\gamma}{\mu} \to 0$, the COLA reduces to an update model similar in OLLGC, in particular when μ is very large, yielding $\mathbf{A}_t^{-1} = \frac{1}{\mu}\Lambda_d^{-1}$, which turns the COLA to be a first-order model with the weight $\mathbf{\Lambda}_d^{-1}$. When $\frac{\gamma}{\mu}$ is very large, the learner weakens the prior graph information $\mathbf{\Lambda}_d^{-1}$ while the learner $\mathbf{w}$ tends to generalize the predicted values. We summarize the COLA in Algorithm 1.

B. Selective Sampling on Graph

Unlike online algorithm that queries all labels, selective sampling has to decide whether to query label or not for each

coming node $\mathbf{x}_t$. If true label $y_t \in \{\pm 1\}$ is queried of, the algorithm can update learner with y_t; otherwise, no action is performed and the learner proceeds next one. Query and update decisions in trial t are defined as binary variables Q_t and Z_t, respectively. When $Q_t = 1$ iif label 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. Thus, its optimal solution can be derived in a form of online learning with query/update decision in each trial, $\mathbf{w}_t = \mathbf{A}_t^{-1}\mathbf{b}_t$, where $\mathbf{A}_t$ and $\mathbf{b}_t$ can turn to be a recursive form,

$$\mathbf{A}_t = \mathbf{A}_{t-1} + Q_t Z_t (1+\gamma)\mathbf{x}_t\mathbf{x}_t^\top, \quad \mathbf{b}_t = \mathbf{b}_{t-1} + Q_t Z_t y_t \mathbf{x}_t. \tag{7}$$

Since $\mathbf{A}_t^{-1}$ is computationally expensive, we derive a non-inverted recursive form using Woodbury formula [2],

$$\begin{aligned}\mathbf{A}_t^{-1} &= (\mathbf{A}_{t-1} + Q_t Z_t (1+\gamma)\mathbf{x}_t\mathbf{x}_t^\top)^{-1} \\ &= \mathbf{A}_{t-1}^{-1} - \frac{Q_t Z_t \mathbf{A}_{t-1}^{-1}\mathbf{x}_t\mathbf{x}_t^\top\mathbf{A}_{t-1}^{-1}}{\frac{1}{1+\gamma} + \mathbf{x}_t^\top\mathbf{A}_{t-1}^{-1}\mathbf{x}_t}\end{aligned} \tag{8}$$

The goal of selective sampling is to achieve few mistakes $\Sigma_t[y_t \neq \hat{y}_t]$ with a small queried number $\Sigma_t Q_t$. To reduce the labeling effort, we propose a novel confidence-based method to decide whether to query in selective sampling algorithm.

C. Conservative Setting

The COLA assumes that all labels are provided, which is not efficient in many real-world applications. To save the labelling cost, we propose a conservative setting of selective sampling. We call the algorithm CGS, Conservative Graph-based Selective Sampling, summarized in Algorithm 2. It maintains two quantities: $\mathbf{A}_t^{-1}$ and $\mathbf{b}_t$ with initial values $\mathbf{A}_0 = \mu\mathbf{\Lambda}_d$ and $\mathbf{b}_0 = \mathbf{0}$. At round t, the algorithm observes a node $\mathbf{x}_t$ and predicts its label with $\mathbf{w}_t = \mathbf{A}_t^{-1}\mathbf{b}_{t-1}$. Then the algorithm decides whether to query the true label. If the label is queried of, the algorithm performs a conservative update, that is, update is invoked by mistake. If an error ($\hat{y}_t \neq y_t$) occurs, the algorithm updates model parameters $\mathbf{A}_t^{-1}$ and $\mathbf{b}_t$ in Eq. (7) and (8); otherwise, the model keeps unchanged and proceeds next nodes with $\mathbf{A}_t^{-1} = \mathbf{A}_{t-1}^{-1}$ and $\mathbf{b}_t = \mathbf{b}_{t-1}$. To select the informative labels, we propose a new query approach for CGS. We begin with additional notations,

$$r_t = \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t. \tag{9}$$

Then an additional notation can be derived,

$$\begin{aligned}\mathbf{x}_t^\top\mathbf{A}_t^{-1}\mathbf{x}_t &= \mathbf{x}_t^\top(\mathbf{A}_{t-1} + (1+\gamma)\mathbf{x}_t\mathbf{x}_t^\top)^{-1}\mathbf{x}_t \\ &\overset{(8)}{=} \mathbf{x}_t^\top(\mathbf{A}_{t-1}^{-1} - \frac{\mathbf{A}_{t-1}^{-1}\mathbf{x}_t\mathbf{x}_t^\top\mathbf{A}_{t-1}^{-1}}{\frac{1}{1+\gamma} + \mathbf{x}_t^\top\mathbf{A}_{t-1}^{-1}\mathbf{x}_t})\mathbf{x}_t \overset{(9)}{=} \frac{r_t}{1+(1+\gamma)r_t}.\end{aligned} \tag{10}$$

Definition 2. *Given an input node $\mathbf{x}_t$ ($t < T$), an algorithm predicts its label with $p_t = \mathbf{x}_t^\top\mathbf{A}_t^{-1}\mathbf{b}_{t-1}$. Given $h > 0$, the actual label is queried with a probability $\frac{h}{h+\max(0,\Theta_t)}$, where Θ_t is a kind of confidence towards its prediction,*

$$\Theta_t = \Theta(p_t, r_t) = |p_t| - \frac{r_t}{2(1+(1+\gamma)r_t)}, \tag{11}$$

where the parameter γ acts as a scaling factor. When γ becomes large, the $\Theta_t \to |p_t|$ is mainly determined by the absolute margin, similar in [7]. When $\gamma = 0$, $\Theta_t = |p_t| - \frac{r_t}{2(1+r_t)}$.

Algorithm 2 CGS: Conservative Graph-based Selective Sampling for Classification

Input: Adjacency matrix $\mathbf{S}$, rank d, regularization parameter μ and γ, and query ratio parameter h.
Output: $\mathbf{w}_T$
Compute $\mathbf{L} = \mathbf{D} - \mathbf{S}$ and $\mathbf{X}$ from $\mathbf{L}$
Initialize: $\mathbf{A}_0 = \mu\mathbf{\Lambda}_d$, $\mathbf{b}_0 = \mathbf{0}$,$\mathbf{w}_0 = \mathbf{0}$
for $t = 1, \ldots, T$ **do**
  Receive $\mathbf{x}_t \in R^d$
  Update $\mathbf{A}_t^{-1} = (\mathbf{A}_{t-1} + (1+\gamma)\mathbf{x}_t\mathbf{x}_t^\top)^{-1}$ as in (8)
  Compute $p_t = \mathbf{x}_t^\top\mathbf{A}_t^{-1}\mathbf{b}_{t-1}$ and $r_t = \mathbf{x}_t^\top\mathbf{A}_{t-1}^{-1}\mathbf{x}_t$
  Calculate $\Theta_t = |p_t| - \frac{r_t}{2(1+(1+\gamma)r_t)}$
  Generate $Q_t \sim \frac{h}{h+\max(0,\Theta_t)}$
  if $Q_t = 1$ **then**
    Query true label y_t and $\hat{y}_t = \text{sign}(p_t)$
    if $y_t \neq \hat{y}_t$ **then**
      Set $Z_t = 1$ ($Z_t = 0$ otherwise)
    end if
  end if
  Update $\mathbf{b}_t = \mathbf{b}_{t-1} + Z_t y_t \mathbf{x}_t$
  Update $\mathbf{A}_t^{-1} = \mathbf{A}_{t-1}^{-1} - \frac{Z_t\mathbf{A}_{t-1}^{-1}\mathbf{x}_t\mathbf{x}_t^\top\mathbf{A}_{t-1}^{-1}}{\frac{1}{1+\gamma}+\mathbf{x}_t^\top\mathbf{A}_{t-1}^{-1}\mathbf{x}_t}$
end for
$\mathbf{w}_T = \mathbf{A}_T^{-1}\mathbf{b}_T$

By tuning the γ, we can achieve a trade-off in the range of these two confidence values.

Intuitively, a query method is effective if it can control the probability of making a mistake whenever the method does not query this label. In the following theorem and corollary, we prove that the CGS learning on these randomly queried labels $\{t : Q_t = 1, Q_t \sim \frac{h}{h+\max(0,\Theta_t)}\}$ achieves a comparable mistake bound with that learns on all the labels.

Let $\mathcal{U}_T = \{t \leq T : Q_t Z_t = 1\}$ be a set of update trials, $h > 0$ is a parameter acting as a scaling factor in the randomized rule, and $\ell(y_t\mathbf{u}^\top\mathbf{x}_t)$ be a hinge loss over $\mathbf{x}_t$. We start with the lemma below.

Lemma 3. *For all $t \geq 1$, a selective sampling predicts with $p_t = \mathbf{x}_t^\top\mathbf{A}_t^{-1}\mathbf{b}_{t-1}$, and make a query and update decision Q_t and Z_t at round t, then given $h > 0$ the following inequality holds for any $\mathbf{u} \in \mathbb{R}^d$,*

$$\begin{aligned}&\sum_{t=1}^{T}\mathbb{E}[Z_t Q_t(|p_t| - \frac{r_t}{2(1+(1+\gamma)r_t)} + h)] \\ &\qquad \leq \frac{h^2}{2}\mathbf{u}^\top\mathbb{E}[\mathbf{A}_{\mathcal{U}_T}]\mathbf{u} + h\mathbb{E}[\sum_{t\in\mathcal{U}_T}\ell(y_t\mathbf{u}^\top\mathbf{x}_t)].\end{aligned} \tag{12}$$

Note that labels are selected randomly. Thus, the expectation occurring in Lemma 3 are w.r.t this randomization.

We leave the proof in the Appendix.

In the proposed randomized query, the mistake trials can be partitioned into two disjoint sets, set $\mathcal{S} = \{t : \frac{h}{h+\max(0,\Theta_t)} < 1\}$ contains instances on which a stochastic query is conduct, while set $\mathcal{D} = \{t : \frac{h}{h+\max(0,\Theta_t)} = 1\}$ includes instances when there is a deterministic query. $\mathcal{M} = \{t : y_t \neq \hat{y}_t\}$ is denoted as the set of mistake trials and let $M = |\mathcal{M}|$.

Theorem 1. *The CGS (in Alg. 2) runs an arbitrary node-label sequence $(\mathbf{x}_1, y_1) \dots (\mathbf{x}_T, y_T)$ ($T \geq 1$) with a query probability of $\frac{h}{h+\max(0,\Theta_t)}$ ($h > 0$), then the following inequality holds for any $\mathbf{u} \in \mathbb{R}^d$,*

$$\mathbb{E}[M] \leq \frac{h}{2}\mathbf{u}^\top \mathbb{E}[A_{\mathcal{U}_T}]\mathbf{u} + \mathbb{E}[\sum_{t\in\mathcal{U}_T} \ell(y_t \mathbf{u}^\top \mathbf{x}_t)] + \frac{1}{2h}\mathbb{E}[\sum_{t\in\mathcal{U}_T\cap\mathcal{D}} \frac{r_t}{1+(1+\gamma)r_t}].$$

The expectation of queried label is upper bound by $\mathbb{E}[|\mathcal{D}| + \sum_{t\in\mathcal{S}} \frac{h}{h+\max(0,\Theta_t)}]$.

Remark. *In this bound, $\mathbf{u}^\top \mathbb{E}[A_{\mathcal{U}_T}]\mathbf{u}$ always lines between $\min_k \lambda_k$ and $\max_k \lambda_k$. For example, $\mathbf{u}^\top \mathbf{A}_{\mathcal{U}_T}\mathbf{u} = \lambda_k$ whenever $\mathbf{u}$ lies in the direction of the eigenvector related to λ_k. Additionally, $\sum_t \frac{r_t}{1+(1+\gamma)r_t} = \sum_t \mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{x}_t \leqslant \log\frac{\det(\mathbf{A}_T)}{\det(\mathbf{A}_0)} = d\log(1+T)$ is substantially smaller than $d\log T$ whenever the spectrum $\mathbf{A}_{\mathcal{U}_T}$ is decreased rapidly. Finally, $\sum_{t\in\mathcal{U}_T} \ell(y_t\mathbf{u}^\top\mathbf{x}_t)$ is a cumulative hinge loss over the update trials.*

Proof: When an error incurs at trial $t \in \mathcal{M}$, the function Θ_t can be positive ($t \in \mathcal{M}\cap\mathcal{S}$) or negative ($t \in \mathcal{M}\cap\mathcal{D}$): In the former subcase, Q_t is random variable with expectation $\mathbb{E}[Q_t] = \frac{h}{h+\Theta_t}$ and based on Lemma 3 we bound,

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

In the later subcase, $\mathbb{E}[Q_t] = 1$. We bound

$$\mathbb{E}[Z_t Q_t(|p_t| - \frac{r_t}{2(1+(1+\gamma)r_t)} + h)] \geq \mathbb{E}[Z_t(-\frac{r_t}{2(1+(1+\gamma)r_t)} + h)] \geq h\mathbb{E}[Z_t] - \frac{1}{2}\mathbb{E}[\frac{r_t}{1+(1+\gamma)r_t}].$$

To summarize,

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}[Z_t Q_t(\Theta_t + h)] \\
&\geq \sum_{t\in\mathcal{M}\cap\mathcal{D}} (h\mathbb{E}[Z_t] - \frac{1}{2}\mathbb{E}[\frac{r_t}{1+(1+\gamma)r_t}]) + \sum_{t\in\mathcal{M}\cap\mathcal{S}} h\mathbb{E}[Z_t] \\
&= h\mathbb{E}[M] - \frac{1}{2}\mathbb{E}[\sum_{t\in\mathcal{U}_T\cap\mathcal{D}} \frac{r_t}{1+(1+\gamma)r_t}].
\end{aligned}$$

Equipped with Eq. (12), we complete our proof. ■

Remark. *The CGS runs on these randomly queried labels achieves, in expectation, a comparable mistake bound with the fully-supervised algorithm OLLGC ([14], Corollary 5) due to two reasons. First, $\sum_{t\in\mathcal{U}_T\cap\mathcal{D}} \frac{r_t}{1+(1+\gamma)r_t} \leq \sum_{t\in\mathcal{U}_T} \mathbf{x}_t A_t^{-1}\mathbf{x}_t \leq \sum_{i=1}^{d} \log(1+\frac{\lambda_i}{\mu}) \leq \log T$ (refer to [14] Lemma 2), where $\lambda_i, i \in [1,d]$) are eigenvalues of matrix $A_{\mathcal{U}_T}$ and T is trial number. Second, $\mathbf{u}^\top A_{\mathcal{U}_T}\mathbf{u} = \sum_{t\in\mathcal{U}_T}(\mathbf{u}^\top\mathbf{x}_t)^2 \leq \mu\|\mathbf{u}\|^2 = \mu\mathbf{f}^\top\mathbf{L}\mathbf{f}$, assuming $\|\mathbf{x}_t\|^2 \leq B$ and $\mu = MB$ (refer to [14] Lemma 3). Generally, the mistake bound of the CGS is comparable with the OLLCG learning on all the labels.*

When the CGS queries all labels, it reduces to the proposed fully-supervised online algorithm COLA.

Corollary 1. *Assuming labels are provided for all the nodes, the CGS querying all labels (i.e. $Q_t = 1$ for all $t \leq T$) reduces to the fully-supervised algorithm COLA, then the following inequality holds for any $\mathbf{u} \in \mathbb{R}^d$,*

$$\mathbb{E}[M] \leq \frac{h}{2}\mathbf{u}^\top \mathbb{E}[A_{\mathcal{U}_T}]\mathbf{u} + \mathbb{E}[\sum_{t\in\mathcal{U}_T} \ell(y_t\mathbf{u}^\top\mathbf{x}_t)] + \frac{1}{2h}\mathbb{E}[\sum_{t\in\mathcal{U}_T} \frac{r_t}{1+(1+\gamma)r_t}].$$

We leave the proof in the Appendix.

Remark. *The theoretical results show that the CGS learning on a fraction of queried labels (refer to Theorem 1) can achieve a mistake bound comparable with the one of CGS learning on all labels (the CGS reduces to the COLA) in expectation, since the term $\mathbb{E}[\sum_{t\in\mathcal{U}_t\cap\mathcal{D}} \frac{r_t}{1+(1+\gamma)r_t}]$ is comparable with $\mathbb{E}[\sum_{t\in\mathcal{U}_t} \frac{r_t}{1+(1+\gamma)r_t}]$. In other word, our selective sampling can achieve a similar performance with its online learning counterpart, with a substantially small number of queried labels. Thus, we conclude that the proposed query approach is effective to control the probability of making a mistake when it decides not to query this label.*

Discussion. Several works combine two quantities to make a randomized query,and can achieve a comparable mistake bound with the fully-supervised methods [7], [9]. Unlike these query competitors, we can tune the parameter γ to leverage a good trade-off for score Θ_t in the range $(|p_t| - \frac{r_t}{2(1+r_t)}, |p_t|)$, which is more flexible to optimize the query rate. In addition, the query method in our algorithm can derive a tiger mistake bound than these query methods [7], [9]. If we consider the impact of the term $-(1+r_t)p_t^2$ to the right side of inequality in Lemma 3 (refer to proof in Appendix), we can derive a tighter mistake bound for the CGS based on our proposed query method,

$$\begin{aligned}
\mathbb{E}[M] \leq& \frac{h}{2}\mathbf{u}^\top \mathbb{E}[A_{\mathcal{U}_T}]\mathbf{u} + \mathbb{E}[\sum_{t\in\mathcal{U}_T} \ell(y_t\mathbf{u}^\top\mathbf{x}_t)] \\
&+ \frac{1}{2h}\mathbb{E}[\sum_{t\in\mathcal{U}_T\cap\mathcal{D}} \frac{r_t}{1+(1+\gamma)r_t}] - \frac{1}{2h}\mathbb{E}[\sum_{t\in\mathcal{U}_T} (1+r_t)p_t^2],
\end{aligned} \tag{13}$$

which is strictly lower than the bounds in (refer to Thm. 2 in [7] and [9]) due to the deduction of $\mathbb{E}[\sum_{t\in\mathcal{U}_T}(1+r_t)p_t^2]$ from the bound. It indicates that a selective sampling based on our query method can achieve a better performance than these query methods in [7], [9] in expectation.

D. Aggressive Setting

The CGS is conservative to update only if an error occurs, thus it discards the labels whenever they are predicted correctly. To take advantage of these labels, we propose an aggressive version of selective sampling on graph. We call our algorithm AGS, the Aggressive Graph-based Selective Sampling, summarized in Algorithm 3. After observing a node $\mathbf{x}_t$ at round t, the AGS predicts its label with $\mathbf{w}_t = \mathbf{A}_t^{-1}\mathbf{b}_{t-1}$ and then queries true label y_t with a probability of $\frac{h}{h+\max(0,\Theta_t)}$. It yields to stochastic query and deterministic query. When stochastic query (i.e. $\frac{h}{h+\max(0,\Theta_t)} < 1$) is issued, it is conservative to update morel when an error occurs ($\hat{y}_t \neq y_t$). While a deterministic query is issued whenever $\frac{h}{h+\max(0,\Theta_t)} = 1$, in this case, we adopt an aggressive learning strategy, that is we update the model even though no error occurs. Note that the

Algorithm 3 AGS: Aggressive Graph-based Selective Sampling for Classification

Input: Adjacency matrix $\mathbf{S}$, rank d, regularization parameter μ and γ, and query ratio parameter h.
Output: $\mathbf{w}_T$
Compute $\mathbf{L} = \mathbf{D} - \mathbf{S}$ and $\mathbf{X}$ from $\mathbf{L}$
Initialize: $\mathbf{A}_0 = \mu\mathbf{\Lambda}_d$, $\mathbf{b}_0 = \mathbf{0}$
for $t = 1, \ldots, T$ **do**
  Receive $\mathbf{x}_t \in R^d$
  Compute $p_t = \mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{b}_{t-1}$ and $r_t = \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t$
  Calculate $\Theta_t = |p_t| - \frac{r_t}{2(1+(1+\gamma)r_t)}$
  if $\frac{h}{h+\max(0,\Theta_t)} = 1$ **then**
    Set $Q_t = 1$ and $Z_t = 1$
    Query label y_t
  end if
  if $\frac{h}{h+\max(0,\Theta_t)} < 1$ **then**
    Generate $Q_t \sim \frac{h}{h+\max(0,\Theta_t)}$
    if $Q_t = 1$ **then**
      Query label y_t
      Set $Z_t = 1$ if $y_t \neq \hat{y}_t$ ($Z_t = 0$ otherwise)
    end if
  end if
  Update $\mathbf{b}_t = \mathbf{b}_{t-1} + Z_t y_t \mathbf{x}_t$
  Update $\mathbf{A}_t^{-1} = \mathbf{A}_{t-1}^{-1} - \frac{Z_t \mathbf{A}_{t-1}^{-1}\mathbf{x}_t\mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}}{\frac{1}{1+\gamma} + \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t}$
end for
$\mathbf{w}_T = \mathbf{A}_T^{-1}\mathbf{b}_T$

AGS is different from the conservative algorithm CGS and SSLGC [14], since we can make use of the correctly predicted labels to update model through an aggressive method. The AGS is different from DAGGER-ridge [9] either, since we propose a more general method to build the model and a better randomized query method to leverage the conservative update with the aggressive one, which can achieve a better performance in expectation. Next we theoretically show the superiority of our aggressive learning algorithm compared to its conservative [14] and fully-supervised counterpart [5].

Besides the stochastic query trials $\mathcal{S}$ and deterministic query trials $\mathcal{D}$, we denote by $\mathcal{V}$ the set of nodes for which there is an aggressive update but not a mistake ($0 < y_t p_t$ and $\Theta_t < 0$) and let $V = \mid \mathcal{V} \mid$. Similar, $\mathcal{U}_t$ is the set of update trials (i.e. $\mathcal{U}_t = \{i \leq t : Z_i Q_i = 1\}$)

Theorem 2. *The algorithm AGS (in Alg. 2) runs on an arbitrary sequential nodes, then given $h > 0$, the following inequality holds for all $\mathbf{u} \in \mathbb{R}^d$,*

$$\mathbb{E}[M] \leq \frac{1}{2}h\mathbf{u}^\top \mathbb{E}(\mathbf{A}_{\mathcal{U}_T})\mathbf{u} + \mathbb{E}[\sum_{t\in\mathcal{U}_T} \ell(y_t \mathbf{u}^\top \mathbf{x}_t)] + \frac{1}{2h}\mathbb{E}[\sum_{t\in\mathcal{D}} \frac{r_t}{1+(1+\gamma)r_t}] - \mathbb{E}[V]. \tag{14}$$

In addition, the expected number of queries is upper bounded by $\mathbb{E}[|\mathcal{D}| + \sum_{t\in\mathcal{S}} \frac{h}{h+\max(0,\Theta_t)}]$.

Proof: When an error occurs $t \in \mathcal{M}$, it either belongs to stochastic query ($t \in \mathcal{M} \cap \mathcal{S}$) or deterministic query ($t \in \mathcal{M} \cap \mathcal{D}$). In the former subcase, Q_t is random variable with $\mathbb{E}[Q_t] = \frac{h}{h+\Theta_t}$ and thus

$$\mathbb{E}[Z_t Q_t(|p_t| - \frac{r_t}{2(1+(1+\gamma)r_t)}) + h] = h\mathbb{E}[\mathbb{Z}_t].$$

In the later subcase, $Q_t = 1$, we bound

$$\mathbb{E}[Z_t Q_t(|p_t| - \frac{r_t}{2(1+(1+\gamma)r_t)}) + h]$$
$$= \mathbb{E}[Z_t(|p_t| - \frac{1}{2}\frac{r_t}{1+(1+\gamma)r_t} + h] \geq h\mathbb{E}[Z_t] - \frac{1}{2}\frac{r_t}{1+(1+\gamma)r_t}.$$

Now we consider the update with correct prediction, that is $y_t p_t \geq 0$ and $\Theta_t \leq 0$. Such cases occur when $t \in \mathcal{D} \cap \mathcal{V}$, thus

$$\mathbb{E}[Z_t Q_t(|p_t| - \frac{1}{2}\frac{r_t}{1+(1+\gamma)r_t}) + h]$$
$$\geq \frac{1}{2}\mathbb{E}[Z_t(-\frac{r_t}{1+(1+\gamma)r_t} + h)] \geq h\mathbb{E}[Z_t] - \frac{1}{2}\frac{r_t}{1+(1+\gamma)r_t}.$$

To summarize,

$$\sum_{t=1}^{T} \mathbb{E}[Z_t Q_t(|p_t| - \frac{r_t}{2(1+(1+\gamma)r_t)}) + h]$$
$$\geq \sum_{t\in\mathcal{M}\cap\mathcal{S}} h\mathbb{E}[Z_t] + \sum_{t\in\mathcal{M}\cap\mathcal{D}} (h\mathbb{E}[Z_t] - \frac{1}{2}\mathbb{E}[\frac{r_t}{1+(1+\gamma)r_t}])$$
$$+ \sum_{t\in\mathcal{V}\cap\mathcal{D}} (h\mathbb{E}[Z_t] - \frac{1}{2}\mathbb{E}[\frac{r_t}{1+(1+\gamma)r_t}])$$
$$\geq h\mathbb{E}[M] + h\mathbb{E}[V] - \frac{1}{2}\sum_{t\in\mathcal{D}} \mathbb{E}[\frac{r_t}{1+(1+\gamma)r_t}].$$

Equipped with Eq. (12), we complete our proof. ■

Remark. *It is obvious that the upper mistake bound of the aggressive algorithm AGS is strictly lower than the conservative algorithm CGS, due to the deduction of $\mathbb{E}[V]$ from the bound. In addition, its mistake bound is also better than a conservative algorithm ([7], Thm. 3), since $\sum_t \frac{r_t}{1+(1+\gamma)r_t} = \sum_t \mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{x}_t \leq \sum_i \log(1+\frac{\lambda_i}{\mu})$, where λ_i is an eigenvalue of matrix $\mathbf{A}_t$. Moreover, this result shows that AGS can achieve a better expected performance than its fully-supervised counterpart COLA, which is proven in Corollary 1. In summary, the theoretical analysis shows that the proposed algorithm AGS, in expectation, can achieve a better performance than its conservative and fully-supervised counterparts, which regards as a theoretical support for the aggressive learning model.*

Remark. *$h > 0$ is a parameter of the algorithm acting as a scaling factor in the randomized rule. The parameter h in the bound (14) is affected by graph structure and order of the inputs. If we would know in advance, by setting a proper choice*

$$h = \sqrt{\frac{\mathbb{E}[\sum_{t\in\mathcal{D}} \frac{r_t}{1+(1+\gamma)r_t}]}{\mathbf{u}^\top \mathbb{E}[A_{\mathcal{U}_T}]\mathbf{u}}},$$

we would minimize the bound and get

$$E[M] \leq \sqrt{\mathbb{E}[\sum_{t\in\mathcal{D}} \frac{r_t}{1+(1+\gamma)r_t}]\mathbf{u}^\top \mathbb{E}[A_{\mathbf{u}}]\mathbf{u}} + \mathbb{E}[\sum_{t\in\mathcal{U}_T} \ell(y_t \mathbf{u}^\top \mathbf{x}_t)] - \mathbb{E}[V]. \tag{15}$$

The first two terms of this optimized bound is an expectation version of the mistake bound for the standard second-order Perceptron algorithm, proved in [5]. As it turn out, our bound in Eq. (15) would be sharper than its fully supervised competitor, since the set of update U_T is formed by a randomized

sampling rule, which is typically smaller than the mistake trials in the deterministic algorithm. This tends to shrink the three terms $\frac{1}{2}h\mathbf{u}^\top A_{\mathcal{U}_T}\mathbf{u}$, $\sum_{t\in\mathcal{U}_T}\ell(y_t\mathbf{u}^\top\mathbf{x}_t)$, *and* $\sum_{t\in\mathcal{D}}\frac{r_t}{1+(1+\gamma)r_t}$, *the main components of our proposed bound.*

Remark. $\sum_{t\in\mathcal{U}_T}\ell(y_t\mathbf{u}^\top\mathbf{x}_t)$ *is a cumulative hinge loss over the update trials. We rewrite the mistake bound as one by which the number of mistakes exceeds a cumulative loss of the best linear model* $\mathbf{u}$ *on update trials,*

$$\begin{aligned}&\mathbb{E}[M]-\inf_{\mathbf{u}}\mathbb{E}[\sum_{t\in\mathcal{U}_T}\ell(y_t\mathbf{u}^\top\mathbf{x}_t)]\\&\leq\frac{1}{2}h\mathbf{u}^\top\mathbb{E}[A_{\mathcal{U}_T}]\mathbf{u}+\frac{1}{2h}\mathbb{E}[\sum_{t\in\mathcal{D}}\frac{r_t}{1+(1+\gamma)r_t}]-\mathbb{E}[V].\end{aligned}\tag{16}$$

Since $\mathbf{u}^\top\mathbb{E}[A_{\mathcal{U}_T}]\mathbf{u}$ *always lines between* $\min_k\lambda_k$ *and* $\max_k\lambda_k$ *and* $\sum_t\frac{r_t}{1+(1+\gamma)r_t}=\sum_t\mathbf{x}_t^\top\mathbf{A}_t^{-1}\mathbf{x}_t\leqslant\log\frac{\det(\mathbf{A}_T)}{\det(\mathbf{A}_0)}=d\log(1+T)$, *our bound in Eq. (16) can achieve a regret of* $O(\log T)$ *with respect to the best linear model, where* T *is the number of the trials.*

Discussion: To further understand the query rule, we compute under what condition a query will be issued aggressively. An aggressive query is issued when $\Theta_t\leq 0$. By solving for $|p_t|$, we get,

$$\Theta_t\leq 0\Rightarrow|p_t|\leq\theta(r_t)=\frac{1}{2}\frac{r_t}{1+(1+\gamma)r_t},\tag{17}$$

If absolute margin value $|p_t|$ is less than $\theta(r_t)$, a query must be issued, while margin value $|p_t|$ is above $\theta(r_t)$, a label is queried with a probability strictly less than 1. And the upper bound of $\theta(r_t)$ increases with r_t. When $r_t=0$, it indicates the algorithm observes same samples many times. An input will be queried only if its margin is zero ($|p_t|\leq\theta(r_t=0)=0$), that is, current hypothesis is unable to predict its label. However, if $r_t=1$ i.e. little knowledge to current input, learner will update its label if its absolute margin $|p_t|$ is no higher than $\theta_\gamma(1)=\frac{1}{4+2\gamma}$, far from the boundary. In particular when $\gamma=0$, the query is issued whenever $|p_t|\leq\theta_{\gamma=0}(1)=0.25$, while γ is large the query is issued if $|p_t|\leq\theta_{\gamma\gg 0}(1)\approx 0$. We empirically evaluate the proposed algorithms in Section V.

E. Low Rank Approximation

Recall that $\mathbf{X}=[\mathbf{v}_1,\ldots,\mathbf{v}_n]^\top$, the graph kernel $\mathbf{f}$ is built exactly, but the time complexity of our algorithm becomes $O(n^2)$, which is computationally expensive for large graphs.

In order to make our online algorithm scalable to large graphs, we propose to choose $\mathbf{X}$ as follows,

$$\hat{\mathbf{X}}=[\mathbf{v}_1,\ldots,\mathbf{v}_d]^\top,\quad\hat{\mathbf{w}}=[\frac{1}{\lambda_1}\mathbf{v}_1,\ldots,\frac{1}{\lambda_d}\mathbf{v}_d]\beta$$

where $d\ll n$. Thus $\hat{\mathbf{f}}=(\sum_{i=1}^d\frac{1}{\lambda_i}\mathbf{v}_i\mathbf{v}_i^\top)\beta$ is a rank-d approximation of the $\mathbf{f}$. And the time complexity of our algorithm is $O(d^2)\ll O(n^2)$.

We analyze the impact of such low-rank approximation on the kernel $\hat{\mathbf{f}}$. We have $\hat{\mathbf{f}}=\hat{\mathbf{L}}^\dagger\beta$, where we denote $\hat{\mathbf{L}}^\dagger=\sum_{i=1}^d\frac{1}{\lambda_i}\mathbf{v}_i\mathbf{v}_i^\top$. Given that $\frac{1}{\lambda_1}\geq\ldots\geq\frac{1}{\lambda_n}$, $\hat{\mathbf{L}}^\dagger$ contains the top d largest eigenvalues of $\mathbf{L}^\dagger$. According to Eckart-Young-Mirsky theorem [11], we claim that $\hat{\mathbf{L}}^\dagger$ is the best rank-d approximation of $\mathbf{L}^\dagger$. Thus, the $\hat{\mathbf{f}}$ is the best rank-d approximation of the $\mathbf{f}$.

Equipped with $\hat{\mathbf{X}}=[\hat{\mathbf{x}}_1,\ldots,\hat{\mathbf{x}}_n]=[\mathbf{v}_1,\ldots,\mathbf{v}_d]^\top\in\mathbb{R}^{d\times n}$, the regularized Laplacian term in Eq. (3) can turn to a sparse form,

$$\mathbf{w}^\top\hat{\mathbf{X}}\mathbf{L}\hat{\mathbf{X}}^\top\mathbf{w}\overset{\text{Lemma 1}}{=}\mathbf{w}^\top\text{diag}(\lambda_1,\ldots,\lambda_d)\mathbf{w}.$$

Thus, a low-rank optimization problem can be rewritten as follows,

$$\arg\min_{\mathbf{w}}\frac{1}{2}\sum_{i=1}^n(\hat{\mathbf{x}}_i^\top\mathbf{w}-y_i)^2+\frac{\mu}{2}\mathbf{w}^\top\hat{\mathbf{\Lambda}}_d\mathbf{w}+\frac{\gamma}{2}\sum_{i=1}^n(\hat{\mathbf{x}}_i^\top\mathbf{w})^2,$$

where $\hat{\mathbf{\Lambda}}_d=\text{diag}(\lambda_1,\ldots,\lambda_d)$ and $\mathbf{w}\in\mathbb{R}^d$.

V. Experimental Results

In this section, we first introduce data sets and experimental evaluation metrics. Then we present the experimental results to evaluate the proposed algorithms.

A. Data Sets and Evaluation Metrics

Data Sets: We introduce four real-world graph data sets used in this paper to evaluate our approaches, which are summarized in Table I.

TABLE I. Description of the data sets

Dataset	*Coauthor*	*Cora*	*IMDB*	*PubMed*
#nodes	1,711	2,485	17,046	19,717
#links	7,507	10,138	993,528	88,651
#classes	4	7	4	3

Coauthor[2] is an undirected co-author graph dataset extracted from *DBLP* database in four areas: "data mining", "machine learning", "databases" and "information retrieval". A total number of 1711 Authors are denoted as nodes while their co-authored relationships are treated as the edges.
Cora[2] contains seven classes of 2485 scientific publications with 5429 citation links. The node is categorized into one of seven classes: "Case based", "Genetic Algorithms", "Neural Networks", "Probabilistic Methods", "Reinforcement Learning", "Rule Learning" and "Theory".
IMDB[3] is an organization that provides up-to-date movie information. The graph is built on co-actor relationship among 17046 movies from four genres: "Romance", "Action", "Animation" and "Thriller".
PubMed[4] contains 19717 scientific publications pertaining to diabetes categorized by one of three types. Publication citation in PubMed consists of 44338 links.
Graphs are supposed to be undirected and connected (the results can be applied to disconnected graphs). If they are directed, we transform them into undirected graphs via $\mathbf{S}\leftarrow\max(\mathbf{S},\mathbf{S}^\top)$. If they are disconnected, we choose the biggest connected subgraph for study.

Evaluation Measures: We evaluate the performance of baselines and our proposed algorithms with two measures:
(i) cumulative error rate, reflecting the prediction accuracy of online learning algorithm;
(ii) number of queried labels, reflecting the label efficiency of

[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/

an algorithm. Note that the smaller the above measures, the better the performance of an algorithm.

Baselines and Parameter Setting: We compare the proposed algorithms with three baselines that are discussed in Section 2. The algorithms we study and their parameter settings are summarized as follows.

GPA[16]: This is the state-of-the-art first order online learning algorithm on graph. There is no required parameter for this algorithm. Note that the Perceptron algorithm is not affected by the step-size.

OLLGC and SSLGC[5] [14]: Both two methods are second-order online learning algorithms on graphs. The OLLGC is a *fully-supervised* algorithm while SSLGC is a *semi-supervised* online algorithm (called selective sampling) that queries a fraction of labels for modeling. All parameters are tuned with grid search on a held-out random shuffle.

COLA, CGS and AGS: COLA is an online learning algorithm while CGS and AGS are selective sampling algorithms. Note that CGS is a *conservative* algorithm that updates model whenever an error occurs, while AGS is an *aggressive* algorithm that can make use of correctly predicted labels to improve model. We set parameter $\gamma = 1$ to avoid overfiiting in all experiments, while the parameter μ is tuned with the grid $\{10^{-3}, 10^{-2}, \ldots, 10\}$ on a held-out random shuffle. The low-rank parameter d is set to 100 since the corresponding performance is good enough while the computational cost is low. For query ratio, we set $h = 0.01$ for Cora and Coauthor, $h = 0.001$ for IMDB and PubMed.

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. In addition, the above algorithms are naturally designed for binary classification, while the data sets have more than two classes. In order to apply the algorithms to those data sets, we use one-vs-rest scheme, which is a standard technique for adapting binary classifiers to the multi-class scenario.

B. Results of Selective Sampling Algorithm

The experimental results are presented in Table II. We found that AGS outperforms all baselines consistently across all four data sets. We also show the results with respect to the round of online learning in Figure 1. In all subfigures, the horizontal-axis represents the rounds of online learning, while the vertical-axis is the cumulative error rate and queried nodes, averaging over 20 times of shuffling order.

We can see the improvement of CGS and AGS over GPA are always significant on every dataset. The reason is that second-order algorithms AGS and CGS updates the linear model $\mathbf{w}_t$ with a covariance matrix $A_t = \sum_t x_t x_t^\top$ which has a spectral structure to correlate with a best linear estimator for observed data instances[5][24]. We also observe that CGS is slightly better than OLLGC and SSLGC, the reason is that CGS employs a query rule that considers both margin (p_t) and uncertainty (r_t), which can query more informative labels to improve the model.

Since AGS uses fewer labels than OLLGC and SSLGC, so intuitively its performance should be no better than OLLGC and SSLGC. However, we can observe that on all graph datasets, AGS always enjoys smaller error rates than SSLGC with much fewer queried nodes, which are crucial to save

[5]we thank authors for kindly sharing the experiential data and code

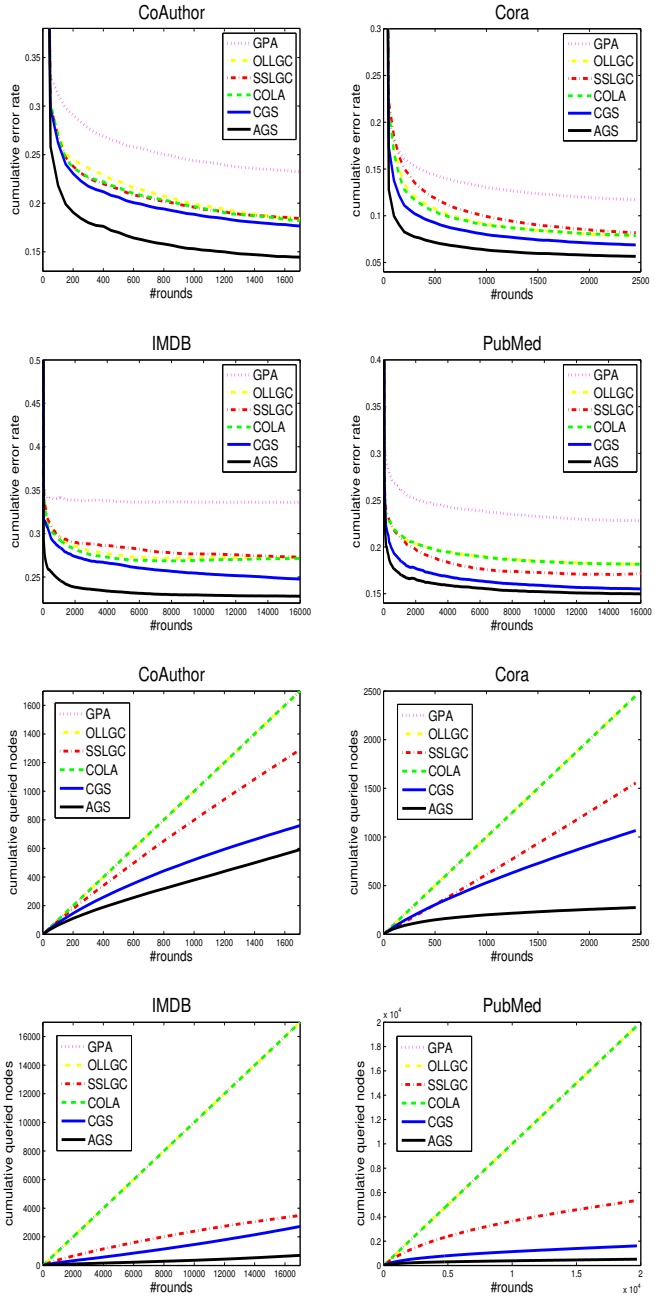

Fig. 1. Cumulative error rate and Number of label query with respective to online learning rounds on four datasets.

labeling efforts while maintain the high quality classification results. The reason is that SSLGC is a mistake-driven algorithm, thus wasting some samples that are correctly predicted, while these samples are properly utilized to improve the model in AGS. Although AGS would aggressively query label in an early learning stage, its model can reach a convergence stage quickly, which further reduces the query number.

C. Sensitivity Study on Low-rank Approximation

The low-rank graph node representation $\mathbf{x} \in \mathbb{R}^d$ is used to build the classification model in our experiments. To analyze the impact of low-rank approximation on our algorithms, we set the rank d using the grid {10, 100, 250, 500, 750, 1000}. We use Cora and Coauthor as a case study as similar observations are obtained on other data sets. The results in

TABLE II. COMPARISON OF SELECTIVE SAMPLING ALGORITHMS. GPA, OLLGC AND COLA ARE ONLINE ALGORITHMS.

Algorithm	*Coauthor*		*Cora*	
	Error rate	# Queried nodes	Error rate	# Queried nodes
GPA	0.2326±0.0048	1711	0.1169±0.0022	2485
OLLGC	0.1839±0.0035	1711	0.0755±0.0013	2485
SSLGC	0.1856±0.0030	1275.30±21.90	0.0832±0.0018	1525.49±19.32
COLA	0.1815±0.0033	1711	0.0760±0.0018	2485
CGS	0.1764±0.0031	762.94±68.03	0.0687±0.0028	1077.6±116.28
AGS	**0.1447±0.0031**	**249.32±37.450**	**0.0566±0.0023**	**274.70±19.390**

Algorithm	*IMDB*		*PubMed*	
	Error rate	# Queried nodes	Error rate	# Queried nodes
GPA	0.3362±0.0025	17046	0.2256±0.0025	19717
OLLGC	0.2737±0.0039	17046	0.1803±0.0013	19717
SSLGC	0.2711±0.0061	3453.98±91.30	0.1721±0.0051	5311.53±178.93
COLA	0.2710±0.0051	17046	0.1795±0.0011	19717
CGS	0.2468±0.0085	2736.3±137.28	0.1540±0.0021	1612.7±115.01
AGS	**0.2293±0.0087**	**709.00±52.290**	**0.1492±0.0021**	**529.95±75.280**

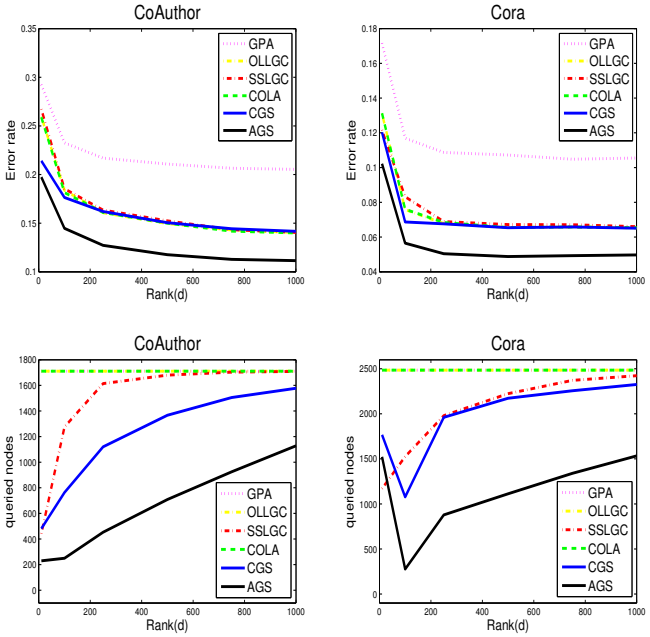

Fig. 2. A case study of low rank impact on performance.

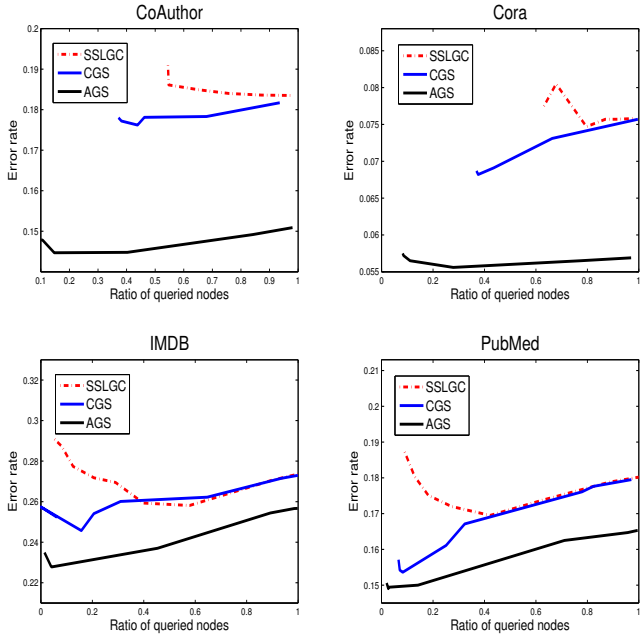

Fig. 3. A comparison among CGS, AGS and SSLGC with respect to different ratios of queried nodes.

Figure 2 show that AGS and CGS achieve better or comparable performance than all baselines consistently under different rank approximation. Obviously with a higher rank, the prediction accuracy becomes better in terms of error rate. However, to achieve a low error rate, algorithms need a high number of queries, which demands a more labeling 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 the algorithms achieve a low error rate while number of queries is small.

D. Study on the Impact of Query Ratio

We study the impact of h w.r.t query ratio in the selective sampling algorithms CGS and AGS. 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 CGS and AGS 20 times under each h. We calculate the average ratio of queried nodes under different values of h. We show comparison results in Figure 3.

We observe that AGS and CGS achieves the better or comparable performance consistently under different ratios of queried nodes. This validates the label-efficiency of our proposed confidence score Θ_t that can adaptively select useful labels to optimize the model. We also observe that the performance of CGS tends to be similar with SSLGC with query ratio increased. The reason is that while algorithms can query more labels, CGS and SSLGC only use error trials to update model, thus some valuable labels are discard since they are correctly predicted. The better performance in AGS demonstrates that these correct predicted labels are effective to improve the model.

VI. CONCLUSION

In this paper, we present better selective sampling algorithms for graph classification. The algorithms are derived from a new online learning linear model on graph and can query labels in both conservative and aggressive ways based on our designed confidence score. We theoretically analyze the mistake bound of proposed algorithms, suggesting that our selective sampling algorithms, with fewer query times, can achieve, on average, a better accuracy than that of its fully supervised setting and other relevant query methods. Additionally, we show the aggressive algorithm can achieve a better mistake bound than the conservative setting, which is considered as a theoretical guarantee for our proposed aggressive method. Finally, our extensive experimental results demonstrate that the proposed algorithms are much better than the state-of-the-arts across multiple real-world graph data.

ACKNOWLEDGMENTS

The authors would like to thank Quanquan Gu for his great help. This study is partially supported by the research grant for the Human-centered Cyber-physical Systems Programme at the ADSC from Singapore's A*STAR.

APPENDIX

A. Proof of Lemma 3

Proof: We first define the objective function $R_t(\mathbf{u})$,

$$R_t(\mathbf{u}) = \sum_{i=1}^{t}(y_i - \mathbf{u}^\top \mathbf{x}_i)^2 + \mu \mathbf{u}^\top \mathbf{\Lambda}_d \mathbf{u} + \gamma \sum_{t=1}^{t}(\mathbf{x}_t^\top \mathbf{u})^2.$$

According to [7] Theorem 2 states,

$$(y_t - p_t)^2 = \inf_{\mathbf{u}} R_t(\mathbf{u}) - \inf_{\mathbf{u}} R_{t-1}(\mathbf{u}) + \mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{x}_t - p_t^2(\mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t).$$

In the setting of selective sampling, if the trial is such that $Z_tQ_t = 0$, then $\mathcal{U}_t = \mathcal{U}_{t-1}$ with no update, which yields $\inf_{\mathbf{u}} R_t(\mathbf{u}) = \inf_{\mathbf{u}} R_{t-1}(\mathbf{u})$. Hence the equality,

$$Z_tQ_t(y_t - p_t)^2 = \inf_{\mathbf{u}} R_t(\mathbf{u}) - \inf_{\mathbf{u}} R_{t-1}(\mathbf{u}) + \frac{Z_tQ_tr_t}{1+r_t} - Z_tQ_tr_tp_t^2,$$

holds for all trials t. We sum over $t = 1, \ldots, T$. Note that $\inf_{\mathbf{u}} R_1(\mathbf{u}) = 0$, Expanding the squares in both sides and performing manipulation, we obtain

$$\begin{aligned}
&\sum_t Z_tQ_t(y_t^2 - 2y_tp_t - \frac{r_t}{1+r_t}) \\
&= \inf_{\mathbf{u}} R_T(\mathbf{u}) - \inf_{\mathbf{u}} R_1(\mathbf{u}) - \sum_t Z_tQ_t(r_t+1)p_t^2 \\
&\leq \sum_{t=1}^{T} Z_tQ_t(y_i - \mathbf{u}^\top \mathbf{x}_t)^2 + \mu \mathbf{u}^\top \mathbf{\Lambda}_d \mathbf{u} + \gamma \sum_{t=1}^{T} Z_tQ_t(\mathbf{x}_t^\top \mathbf{u})^2 \\
&\leq \mathbf{u}^\top(\mu \mathbf{\Lambda}_d + (1+\gamma)\sum_{t=1}^{T} Z_tQ_t\mathbf{x}_t\mathbf{x}_t^\top)\mathbf{u} + \sum_t Z_tQ_t(y_t^2 - 2y_t\mathbf{u}^\top \mathbf{x}_t),
\end{aligned}$$

holding for any $\mathbf{u} \in \mathbb{R}^d$. We drop the term $-Z_tQ_t(r_t+1)p_t^2$, which is negative ($r_t \geq 0$ and $p_t^2 \geq 0$) and do not affect the upper bound. Note that we reconsider the impact of $-Z_tQ_t(r_t+1)p_t^2$ in Eq. (13) to compare other query methods.

Since $\mathbf{u}$ is a random variable, we use $h\mathbf{u}$ to replace $\mathbf{u}$ where $h > 0$. Using inequality $1 - x \leq \max\{1-x, 0\}$ yields

$$hZ_tQ_t - hZ_tQ_ty_t\mathbf{u}^\top \mathbf{x}_t \leq hZ_tQ_t\ell(y_t\mathbf{u}^\top \mathbf{x}_t).$$

Observing $-y_tp_t = |p_t|$ whenever an error $Z_tQ_t = 1$, we simplify with the notations $\mathbf{A}_t$ and Θ_t,

$$\sum_{t=1}^{T}\mathbb{E}[Z_tQ_t(\Theta_t + h)] \leq \frac{h^2}{2}\mathbf{u}^\top \mathbb{E}[\mathbf{A}_{\mathcal{U}_T}]\mathbf{u} + h\mathbb{E}[\sum_{t\in\mathcal{U}_T}\ell(y_t\mathbf{u}^\top \mathbf{x}_t)].$$

■

B. Proof of Corollary 1

Proof: We have $\mathbb{E}[Q_t] = 1$ due to $Q_t = 1$ for all trials. We invoke an update when an mistake occurs, thus we have $M = \sum_t Z_t$. We bound

$$\begin{aligned}
\sum_{t=1}^{T}\mathbb{E}[Q_t]\mathbb{E}[Z_t(\Theta_t + h)] &\geq \frac{1}{2}\sum_{t=1}^{T}\mathbb{E}[Z_t(-\frac{r_t}{1+(1+\gamma)r_t} + 2h)] \\
&\geq h\mathbb{E}[M] - \frac{1}{2}\sum_{t\in\mathcal{U}_T}\mathbb{E}[\frac{r_t}{1+(1+\gamma)r_t}].
\end{aligned}$$

Equipped with Eq. (12), we complete our proof. ■

REFERENCES

[1] K. S. Azoury and M. K. Warmuth, "Relative loss bounds for on-line density estimation with the exponential family of distributions," *Machine Learning*, vol. 43, no. 3, pp. 211–246, 2001.

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

[3] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *JMLR*, vol. 7, pp. 2399–2434, 2006.

[4] G. Cavallanti, N. Cesa-Bianchi, and C. Gentile, "Learning noisy linear classifiers via adaptive and selective sampling," *Machine learning*, vol. 83, no. 1, pp. 71–102, 2011.

[5] N. Cesa-Bianchi, A. Conconi, and C. Gentile, "A second-order perceptron algorithm," *SIAM Journal on Computing*, vol. 34, no. 3, pp. 640–668, 2005.

[6] N. Cesa-Bianchi, C. Gentile, and F. Orabona, "Robust bounds for classification via selective sampling," in *ICML*, 2009, pp. 121–128.

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

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

[9] K. Crammer, "Doubly aggressive selective sampling algorithms for classification," in *AIStat*, 2014, pp. 140–148.

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

[11] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[12] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Machine learning*, vol. 28, no. 2-3, pp. 133–168, 1997.

[13] A. B. Goldberg, X. Zhu, A. Furger, and J.-M. Xu, "Oasis: Online active semi-supervised learning," in *AAAI*, 2011.

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

[15] Y. Guo and R. Greiner, "Optimistic active learning using mutual information," in *IJCAI*, 2007, pp. 823–829.

[16] M. Herbster and M. Pontil, "Prediction on a graph with a perceptron," in *NIPS*, 2006, pp. 577–584.

[17] M. Herbster, M. Pontil, and L. Wainer, "Online learning over graphs," in *ICML*, 2005, pp. 305–312.

[18] S. C. H. Hoi, J. Wang, and P. Zhao, "LIBOL: a library for online learning algorithms," *JMLR*, vol. 15, no. 1, pp. 495–499, 2014.

[19] D. Kushnir, "Active-transductive learning with label-adapted kernels," in *ACM SIGKDD*, 2014, pp. 462–471.

[20] J. Lu, P. Zhao, and S. C. H. Hoi, "Online passive aggressive active learning and its applications," in *ACML 2014, Nha Trang City, Vietnam, November 26-28, 2014.*, 2014.

[21] F. Orabona and N. Cesa-Bianchi, "Better algorithms for selective sampling," in *ICML*, 2011, pp. 433–440.

[22] N. Slonim, E. Yom-Tov, and K. Crammer, "Active online classification via information maximization," in *IJCAI*, 2011, pp. 1498–1504.

[23] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *Learning theory and kernel machines*, 2003, pp. 144–158.

[24] J. Wang, P. Zhao, and S. C. Hoi, "Exact soft confidence-weighted learning," in *ICML*, 2012, pp. 121–128.

[25] Z. Yang, J. Tang, and Y. Zhang, "Active learning for streaming networked data," in *CIKM*, 2014, pp. 1129–1138.

[26] P. Zhao and S. C. H. Hoi, "Cost-sensitive online active learning with application to malicious URL detection," in *SIGKDD 2013, Chicago, IL, USA, August 11-14, 2013*, 2013, pp. 919–927.

[27] P. Zhao, S. C. H. Hoi, and R. Jin, "Double updating online learning," *JMLR*, vol. 12, pp. 1587–1615, 2011.

[28] P. Zhao, S. C. H. Hoi, R. Jin, and T. Yang, "Online AUC maximization," in *ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, 2011, pp. 233–240.