
Active Learning on Attributed Graphs via Graph Cognizant Logistic Regression and Preemptive Query Generation

Florence Regol^{†1} Soumyasundar Pal¹ Yingxue Zhang² Mark Coates¹

Abstract

Node classification in attributed graphs is an important task in multiple practical settings, but it can often be difficult or expensive to obtain labels. Active learning can improve the achieved classification performance for a given budget on the number of queried labels. The best existing methods are based on graph neural networks, but they often perform poorly unless a sizeable validation set of labelled nodes is available in order to choose good hyperparameters. We propose a novel graph-based active learning algorithm for the task of node classification in attributed graphs; our algorithm uses graph cognizant logistic regression, equivalent to a linearized graph-convolutional neural network (GCN), for the prediction phase and maximizes the expected error reduction in the query phase. To reduce the delay experienced by a labeller interacting with the system, we derive a preemptive querying system that calculates a new query during the labelling process, and to address the setting where learning starts with almost no labelled data, we also develop a hybrid algorithm that performs adaptive model averaging of label propagation and linearized GCN inference. We conduct experiments on five public benchmark datasets, demonstrating a significant improvement over state-of-the-art approaches and illustrate the practical value of the method by applying it to a private microwave link network dataset.

¹Department of Electrical and Computer Engineering, McGill University, Montréal, QC, Canada. ² Huawei Noah's Ark Lab, Montréal Research Center, Montréal, QC, Canada. Correspondence to: Florence Regol <florence.robert-regol@mail.mcgill.ca>.

Proceedings of the 37th International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

[†]Work done as intern at Huawei Noah's Ark Lab, Montréal Research Center.

1. Introduction

In many classification tasks there are explicit or implicit relationships between the data points that need to be classified. One can represent such data using a graph, where an edge between two nodes (data points) indicates the presence of a relationship. The resultant task of node classification has attracted significant attention from the graph-learning research community, and numerous graph learning architectures have been developed that yield impressive performance, especially in semi-supervised settings (Defferrard et al., 2016; Kipf & Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018; Zhuang & Ma, 2018; Gao et al., 2018a; Liu et al., 2019). In such cases, knowledge of the graph topology can compensate for scarcity of labelled data.

In practice, the semi-supervised classification task often arises in scenarios where it is challenging or expensive to obtain labels. If we have the opportunity to decide which nodes to query, then we should try to select the most informative nodes that lead to the best classification accuracy. This is, in a nutshell, the goal of active learning; as we acquire labels, we make decisions about which label to query next based on what we have learned. This is important in applications such as medical imaging, where generating labels requires considerable valuable time from domain experts (Hoi et al., 2006; Gal et al., 2017; Kurzendorfer et al., 2017). The development of active learning algorithms for node classification in graphs can be motivated by applications of graph convolutional neural networks (GNNs) in the medical field. For example, in (Parisot et al., 2018), GNNs are used to classify brain scan images, with the goal of predicting disease outcomes or detecting the presence of a disorder. Although we may have access to many brain scans and can specify relationships between them (thus building a graph), obtaining labels for them is expensive because it requires attention from medical experts.

The early research that applies active learning to graph data mainly focuses on the non-attributed graph setting (Zhu et al., 2003b; Ji & Han, 2012; Ma et al., 2013). We focus on node classification for attributed graphs, so the recently proposed GNN-based methods (Cai et al., 2017; Gao et al., 2018b) are more aligned with the task we address. The results reported in these works are usually based on GNNs

with hyperparameters that have been optimized using a large labelled validation set. This is an unrealistic setting; if we have access to such a large amount of labeled data, we should use much more of it to train the classifier. As we illustrate in the experiments in Section 5, if hyperparameters are not optimized, but are chosen randomly from a reasonable range of candidate values, the performance of the GNN-based active learning methods deteriorates dramatically.

In this work, we aim to address the limitations of the GNN methods. We propose an algorithm that is based on the Expected Error Minimization (EEM) framework (Settles, 2009). In this framework, we select the query that minimizes the expected classification error according to our current model. We use the simplified graph convolution (SGC) (Wu et al., 2019), a graph-cognizant logistic regression, as a predictive model for the labels. This model, which can be derived as a linearization of the graph convolutional neural network (Wu et al., 2019), performs much better when there is limited data compared to a GNN with suboptimal hyperparameters, and achieves competitive accuracy as the number of labels increases.

Most active learning techniques involve initial training of a model and then an iterative process of (i) identifying the best query by some criterion (the core step of active learning); (ii) obtaining the label from an oracle; and (iii) updating the model. In an interactive application, this can lead to a delay if the query generation of step (i) is not extremely fast. Although it is principled and competitive to other approaches, the EEM algorithm does have the disadvantage of an increased computational overhead. However, we note that a delay will also be introduced at step (ii); human labelling can take seconds (document categorization (Settles, 2009)) to minutes (cancer diagnosis from skin lesion images (Gal et al., 2017), MRI tumour segmentation (Kurzendorfer et al., 2017), or fault detection in microwave link networks). With this in mind, the interactive delay can be reduced or even eliminated if the model update and query identification steps can be started and completed while the labelling is conducted. We develop such a preemptive strategy, based on a prediction of the labelling from the oracle.

In summary, our paper makes the following contributions: (i) we propose a practical approach for active learning in graphs that does not have the unrealistic requirement of a validation set for hyperparameter tuning; (ii) we extend the proposed approach to introduce preemptive query generation in order to reduce or eliminate the delay experienced by a labeller during interaction with the system; (iii) we derive bounds on the error in risk evaluation associated with the preemptive prediction; (iv) we analyze performance on five public benchmark datasets and show a significant improvement compared to state-of-the-art GNN active learning

methods (and label propagation strategies); (v) we illustrate the practical benefit of our method by demonstrating its application to a private, commercial dataset collected for the task of identifying faulty links in a microwave link network.

2. Related Research

2.1. Active learning on non-attributed graphs

Many methods for active learning on graphs without node or edge features are based on the idea of propagating label information across the graph, and we hence refer to them throughout the paper as label-propagation methods. The most successful techniques are all based on the Binary Random Markov Field (BRMF) model. The model allows one to define a posterior on the unknown labels conditioned on the graph topology and the observed node labels. This model provides an effective mechanism for representing smoothness of labels with respect to the graph, but evaluating the posterior is a combinatorial problem. As a result, researchers have introduced relaxations or approximation strategies. (Zhu et al., 2003a) relax the BMRF to a Gaussian Random Field (GRF) model, and (Zhu et al., 2003b; Ji & Han, 2012) and subsequently (Ma et al., 2013) also employ this model to derive active learning methods. More recently, (Berberidis & Giannakis, 2018) have applied the expected change maximization strategy to a GRF model. (Jun & Nowak, 2016) takes another approach by proposing a two-step approximation (TSA) of the intractable combinatorial problem rather than relaxing the BMRF model.

These strategies offer the advantage of being principled methods that directly target the quantity we want to optimize, but label propagation-based models cannot take into account node features and consequently must rely on strong assumptions regarding the relationships between the graph topology and the data. Most label propagation methods struggle if the graph is not connected and do not usually translate well to an inductive setting, because query decisions rely on the knowledge of the complete graph topology.

2.2. Graph neural network methods for active learning

(Cai et al., 2017) leverage the output of a Graph Convolution Network (GCN) (Kipf & Welling, 2017) to design active learning metrics. Their method is to alternate during the training of the GCN between adding one node to the training set and performing one epoch of training. Selection of the query node is based on a score that is a weighted mixture of three metrics covering different active learning strategies: an uncertainty metric, a density-based metric and a graph centrality metric. The uncertainty metric is obtained by taking the entropy of the softmax output given by the current GCN model. The density metric is based on the GCN node embeddings; the embeddings are clustered

and the distance between each node’s embedding and the centre of its cluster is computed. A more central embedding indicates a more representative node. The graph centrality metric is independent of the GCN and only relies on the position of the node in the graph. The weights change as more nodes are added to the labelled set, in order to reflect the increased confidence in the two metrics that are derived from the output of the GCN. The weight adaptation schedule in (Cai et al., 2017) is fixed; (Gao et al., 2018a) propose an alternative multi-armed bandit algorithm that learns how to balance the contributions of the different metrics. They argue that this mechanism can better adapt to the varying natures of different datasets.

3. Problem Setting

3.1. Pool-based Formulation

We consider the problem of active learning on an attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ for node classification using feature matrix \mathbf{X} and labels \mathbf{Y} . The nodes are partitioned into two sets: a small initial labelled set \mathcal{L}_0 from with node labels $\mathbf{Y}_{\mathcal{L}_0}$ ($|\mathcal{L}_0| \ll |\mathcal{V}|$), and a set \mathcal{U}_0 consisting of the remaining unlabelled nodes. The algorithm is given a budget of b nodes that it can query from \mathcal{U}_0 to augment $\mathbf{Y}_{\mathcal{L}_0}$. We denote by \mathcal{L}_t and \mathcal{U}_t the sets of labelled and unlabelled nodes, respectively, after t nodes have been added to the initial labelled set. The pool-based active learning formulation that we are considering consists of three phases:

1. **Prediction Step** : \mathbf{X} , \mathcal{G} and the current node labels, \mathcal{L}_t , are used to infer the labels of the nodes \mathcal{V} .
2. **Query Step** : Until the budget is exhausted, select a node $q_t^* \in \mathcal{U}_t$ to query and to add to the labelled set \mathcal{L}_t .
3. **Labeling Step** : The oracle takes time Δ to label q_t^* . We update the sets: $\mathcal{U}_{t+1} = \mathcal{U}_t \setminus \{q_t^*\}$, $\mathcal{L}_{t+1} = \mathcal{L}_t \cup \{q_t^*\}$.

The goal is to select the best node q_t^* to append to \mathcal{L}_t at each iteration t , in order to optimize the prediction performance throughout the query process. We are not only interested in the end result after exhausting the query budget, but also in how quickly we can increase accuracy. Acquiring labels is presumed to be expensive, so a solution that reaches competitive performance with fewer nodes is desired.

In addition to the *transductive* setting outlined above, where we know the entire graph and all attributes, we also consider an *inductive* setting, where our goal is to maximize performance over an additional set of nodes \mathcal{T} ; we know that these are connected to the graph in some fashion, but we cannot query them and we do not know the edges or features during the active learning process.

3.2. Reducing interaction delay

With the phases outlined above, the active learning algorithm stalls while waiting for the oracle to label q_t^* at the third phase and then the oracle must wait while the algorithm computes the best subsequent query node q_{t+1}^* . This is inefficient and, for a human oracle, frustrating. We can address this by requiring the active learning algorithm to identify the query node q_t^* using the label set \mathcal{L}_{t-1} . If the labelling time and the query computation time are similar, then neither the oracle nor the algorithm stalls for long. While the oracle is generating the label for q_t^* , this *preemptive* active learning algorithm identifies in parallel the best query node q_{t+1}^* using \mathcal{L}_t . Figure 1 compares the timelines of the standard single-query active learning procedure (query generation algorithm waits for the oracle and vice versa) with the preemptive strategy where labelling and query generation are performed in parallel.

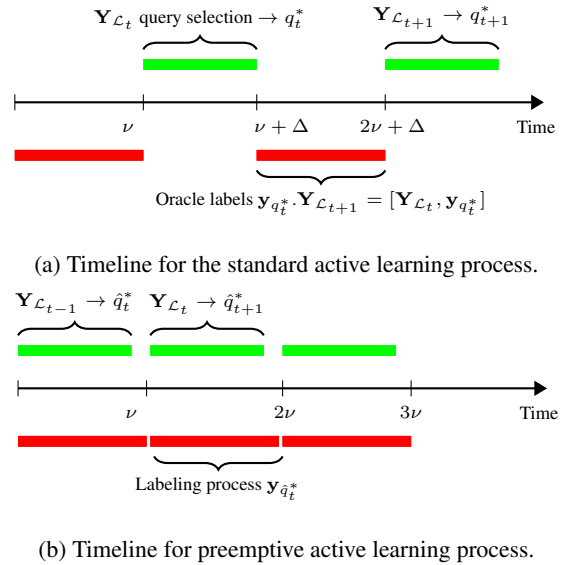


Figure 1. A comparison of the timelines of the standard single-query active learning process and the proposed preemptive process.

4. Methodology

4.1. Expected error Minimization (EEM)

An active learning algorithm based on error reduction selects the query q^* to minimize the expected error. For a classification task, the zero-one error is a suitable choice. Denoting by \mathcal{U}_t^{-q} the set of unlabelled nodes after t iterations of active learning with node q removed, i.e., $\mathcal{U}_t^{-q} \triangleq \mathcal{U}_t \setminus \{q\}$. The labels associated with this set are $\mathbf{Y}_{\mathcal{U}_t \setminus \{q\}}$. Following (Jun & Nowak, 2016), we can define $R_{|\mathbf{Y}_{\mathcal{L}_t}|}^{+q}$, the risk of adding node $q \in \mathcal{U}_t$ given the current known label set \mathcal{L}_t , as:

$$R_{|\mathbf{Y}_{\mathcal{L}_t}|}^{+q} \triangleq E_{\mathbf{y}_q} \left[E_{\mathbf{Y}_{\mathcal{U}_t^{-q}}} \left[\frac{1}{|\mathcal{U}_t^{-q}|} \sum_{i \in \mathcal{U}_t^{-q}} \mathbb{1}[\hat{y}_i \neq \mathbf{y}_i | \mathbf{y}_q, \mathbf{Y}_{\mathcal{L}_t}] \right] \right] \quad (1)$$

Here \hat{y}_i is the label prediction at node i . We thus calculate expected error by summing error probabilities over the unlabelled set, minus the node q we are considering. Define $\varphi_{i,k,\mathbf{Y}_{\mathcal{L}_t}}^{+q} \triangleq \left(1 - \max_{k' \in K} p(y_i = k' | \mathbf{Y}_{\mathcal{L}_t}, y_q = k)\right)$, where K is the set of classes. If the query node \mathbf{y}_q has label k , then $\varphi_{i,k,\mathbf{Y}_{\mathcal{L}_t}}^{+q}$ represents the probability of making an error in the prediction \hat{y}_i of the label of node i . If we can compute the distribution $p(\mathbf{y}|\cdot)$, we can evaluate the risk of querying q :

$$R_{|\mathbf{Y}_{\mathcal{L}_t}}^{+q} = \frac{1}{|\mathcal{U}_t^{-q}|} \sum_{k \in K} \sum_{i \in \mathcal{U}_t^{-q}} \varphi_{i,k,\mathbf{Y}_{\mathcal{L}_t}}^{+q} p(y_q = k | \mathbf{Y}_{\mathcal{L}_t}) \quad (2)$$

The query algorithm selects the risk-minimizing node q_t^* :

$$q_t^* = \arg \min_{q \in \mathcal{U}_t} R_{|\mathbf{Y}_{\mathcal{L}_t}}^{+q} \quad (3)$$

It remains to define the probabilistic model $p(\mathbf{y}|\cdot)$.

4.2. Graph-cognizant logistic regression

We propose to use a graph-cognizant logistic regression model to obtain $p(\mathbf{y}|\cdot)$. Such a model was introduced by (Wu et al., 2019), where the SGC is derived as a simplified (linearized) version of the graph convolutional network of (Kipf & Welling, 2017). (Wu et al., 2019) showed that the simplified model can achieve competitive performance for a significantly lower computational cost. In the EEM approach to active learning, we must learn a new model for every potential query node, so it is essential that the computational cost is relatively low. The SGC meets our requirements: its computational requirements are moderate and it takes into account the graph structure and node features.

For a graph with adjacency matrix \mathbf{A} , let $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, \mathbf{D} be the degree matrix, and $\tilde{\mathbf{D}} = \mathbf{D} + \mathbf{I}$. We then define $\mathbf{S} \triangleq \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$. This can be interpreted as a degree normalized symmetrized adjacency matrix (after self-loops have been added by the identity matrix \mathbf{I}). The prediction model has the form:

$$\hat{\mathbf{Y}} = \sigma(\mathbf{S}^\ell \mathbf{X} \mathbf{W}). \quad (4)$$

We define $\tilde{\mathbf{X}}_{\mathcal{G}} \triangleq \mathbf{S}^\ell \mathbf{X}$; this can be interpreted as graph-based preprocessing of node features. The parameter ℓ controls the number of hops that are considered when generating the final node representation. Usually using a 2-hop ($\ell = 2$) neighborhood yields good results.

4.3. Graph EEM (GEEM)

Using the SGC model, we can compute a risk for each query node. At each step t , we use the current known labels $\mathbf{Y}_{\mathcal{L}_t}$ to find the weights $\mathbf{W}_{\mathbf{Y}_{\mathcal{L}_t}}$ by minimizing the error for predictions $\hat{\mathbf{Y}}_{\mathcal{L}_t} = \sigma(\tilde{\mathbf{X}}_{\mathcal{L}_t} \mathbf{W}_{\mathbf{Y}_{\mathcal{L}_t}})$. We use a standard iterative algorithm for maximum likelihood logistic regression

with L2 regularization (e.g., the scikit-learn liblinear solver). We can then compute $p(\mathbf{y}_q = k | \mathbf{Y}_{\mathcal{L}_t}) = \sigma(\tilde{\mathbf{x}}_q \mathbf{W}_{\mathbf{Y}_{\mathcal{L}_t}})^{(k)}$, where the index (k) indicates that we extract the k -th element of the vector. Then for each candidate node q , for each possible class k , we solve:

$$\hat{\mathbf{Y}}_{\mathcal{L}_t, +q, \mathbf{y}_k} = \sigma(\tilde{\mathbf{x}}_{\mathcal{L}_t, +q, \mathbf{y}_k} \mathbf{W}_{\mathbf{Y}_{\mathcal{L}_t, +q, \mathbf{y}_k}}). \quad (5)$$

Here the notation $+q, \mathbf{y}_k$ indicates that we are adding node q to the labelled set and assigning it label \mathbf{y}_k . For the adopted model, $\varphi_{i,k,\mathbf{Y}_{\mathcal{L}_t}}^{+q} = \left(1 - \max_{k' \in K} \sigma(\tilde{\mathbf{x}}_i \mathbf{W}_{\mathbf{Y}_{\mathcal{L}_t, +q, \mathbf{y}_k}})^{(k')}\right)$. The node to query is then the one that minimizes the risk:

$$q_t^* = \arg \min_{q \in \mathcal{U}_t} \frac{1}{|\mathcal{U}_t^{-q}|} \sum_{k \in K} \sum_{i \in \mathcal{U}_t^{-q}} \varphi_{i,k,\mathbf{Y}_{\mathcal{L}_t}}^{+q} \sigma(\tilde{\mathbf{x}}_q \mathbf{W}_{\mathbf{Y}_{\mathcal{L}_t}})^{(k)} \quad (6)$$

From this formulation, we can see that we first have to evaluate $p(\mathbf{y}|\mathbf{Y}_{\mathcal{L}_t})$, then calculate $p(\mathbf{y}|\mathbf{Y}_{\mathcal{L}_t}, \mathbf{y}_q = k)$ for each of the $|K|$ potential augmented labelled sets $\mathcal{L}^{+q, \mathbf{y}_k}$. This implies that we have a computational complexity of $\mathcal{O}(|\mathcal{U}||K|M)$, where M represents the complexity associated with training the model. For logistic regression, this is the overhead involved in learning the weights $\mathbf{W}_{\mathbf{Y}_{\mathcal{L}_t}}$. If the evaluation of $p(\mathbf{y}|\cdot)$ is computationally expensive, then the time required to select a query node can become prohibitive. This is a common disadvantage of the expected reduction strategies (Settles, 2009). It then becomes apparent why using the linearized version of the GCN is important.

The proposed algorithm requires the choice of very few hyperparameters (only the number of hops ℓ and logistic regression parameters). This contrasts with the active learning approaches based on graph neural networks, where there are multiple hyperparameters that must be selected, and suboptimal choices can have a major impact on performance.

4.4. Preemptive Query (PreGEEM)

In many practical active learning scenarios, labelling is performed by a human, and it is often time-consuming; labelling a single data point (node) can take tens of seconds or minutes. It is desirable to have the next query identified as soon as the labeller has completed the labelling task. With the EEM algorithm formulated above, this is impossible, because the query identification in (6) uses the label associated with the previous query node.

In this section, we outline an alternative approach that performs pre-emptive query calculation, using the labelling time to identify the next node to query. Instead of waiting for the oracle to label q_{t-1}^* to start the identification of q_t^* during iteration t , we propose to approximate the risk before knowing $\mathbf{y}_{q_{t-1}^*}$. The direct approach is to replace the risk $R_{|\mathbf{Y}_{\mathcal{L}_t}}^{+q_t}$ with the expectation over the possible values

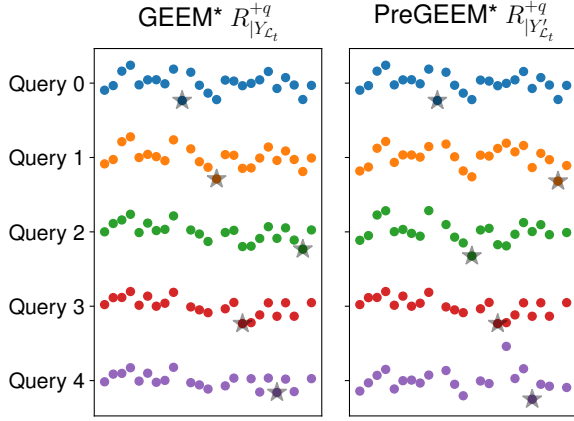


Figure 2. Risk comparison for GEEM vs PreGEEM. This follows the risk computations for 25 nodes in cora dataset for one trial. The black star indicates which node was selected, (following the algorithm, it is the one with the lowest expected risk.)

of $\mathbf{y}_{q_{t-1}^*}$, but this increases the computational complexity by a factor of $|K|$, which is highly undesirable. To avoid this penalty, we further approximate $R_{|\mathbf{Y}_{\mathcal{L}_{t-1}}, q_{t-1}^*}^{+q}$ using the value of risk for the label at the mode of $p(\mathbf{y}_{q_{t-1}^*} | \mathbf{Y}_{\mathcal{L}_{t-1}})$. Effectively, we use the predicted label $\hat{\mathbf{y}}_{q_{t-1}^*}$ of the previous model $p(\cdot | \mathbf{Y}_{\mathcal{L}_{t-1}})$ to form an augmented labelled set $\mathbf{Y}'_{\mathcal{L}_t} = \{\mathbf{Y}_{\mathcal{L}_{t-1}} \cup \{\hat{\mathbf{y}}_{q_{t-1}^*}\}\}$ and define an approximate risk:

$$\hat{R}_{|\mathbf{Y}'_{\mathcal{L}_t}}^{+q} \triangleq E_{\mathbf{y}_q} \left[E_{\mathbf{Y}_{\mathcal{U}_t^q}} \left[\frac{1}{|\mathcal{U}_t^q|} \sum_{i \in \mathcal{U}_t^q} \mathbb{1}[\hat{\mathbf{y}}_i \neq \mathbf{y}_i | \mathbf{y}_q, \mathbf{Y}_{\mathcal{L}_t}] \right] \right]. \quad (7)$$

The query node is then $\hat{q}_t^* = \arg \min_{q \in \mathcal{U}_t} \hat{R}_{|\mathbf{Y}'_{\mathcal{L}_t}}^{+q}$. We call this new approach the Preemptive Graph EEM (PreGEEM).

Figure 2 compares the evolution of the two active learning algorithms GEEM and PreGEEM for a small subset of nodes to illustrate how using the approximated risk can impact the query process. We see that the evaluated risks are similar, and although the ordering of query nodes differs, after five steps the same nodes have been selected by both algorithms.

4.5. Bounds on the PreGEEM Risk Error

We now present bounds on the risk estimation error that can arise by using a predicted label. We focus on the one-step error $|R_{|\mathbf{Y}'_{\mathcal{L}_t}}^{+q} - R_{|\mathbf{Y}_{\mathcal{L}_t}}^{+q}|$, where the labelled sets differ for only one label. For clarity, we first derive a bound for the binary classification task. We then state the more general bound for multiclass classification. Complete proofs are provided in the supplementary material.

The following proposition, which follows straightforwardly from Theorem 1 and Lemma 1 in (Sivan et al., 2019), bounds the difference in prediction values for two label sets that

differ for m data points. The bound is expressed in terms of the regularization parameter λ and the graph pre-processed feature vectors $\tilde{\mathbf{x}}_i$.

Proposition 1. For weights \mathbf{w}_1 and \mathbf{w}_2 derived by L2-penalized logistic regression to two datasets with common feature vectors but label sets differing for m vectors indexed by $\{i_k\}, k = 1, \dots, m$, define $\eta \triangleq \frac{1}{2\lambda} \sum_{k=1}^m \|\tilde{\mathbf{x}}_{i_k}\|$ and $b_{\pm\eta}(\mathbf{w}_2, i) \triangleq \sigma(\mathbf{w}_2^\top \tilde{\mathbf{x}}_i) - \sigma(\mathbf{w}_2^\top \tilde{\mathbf{x}}_i \pm 2\eta \|\tilde{\mathbf{x}}_i\|)$. For any $\tilde{\mathbf{x}}_i$, $|\sigma(\mathbf{w}_1^\top \tilde{\mathbf{x}}_i) - \sigma(\mathbf{w}_2^\top \tilde{\mathbf{x}}_i)| \leq \max(|b_\eta(\mathbf{w}_2, i)|, |b_{-\eta}(\mathbf{w}_2, i)|)$.

The following result characterizes the risk error when we perform L2-regularized binary logistic regression on $\{\mathbf{Y}'_{\mathcal{L}_t} \cup \{\mathbf{y}_q = k\}\}$ to derive weights $\mathbf{w}_{2,k}$ for $k \in \{0, 1\}$.

Define $\eta_q \triangleq \frac{1}{2\lambda} (\|\tilde{\mathbf{x}}_q\| + \|\tilde{\mathbf{x}}_{q_{t-1}^*}\|)$, and $\tilde{b}(\eta_q, \mathbf{w}_2, i) = \max_{k \in \{0, 1\}} \max(|b_{+\eta_q}(\mathbf{w}_{2,k}, i)|, |b_{-\eta_q}(\mathbf{w}_{2,k}, i)|)$.

Theorem 4.1. The risk error arising from applying binary L2-regularized logistic regression with regularization parameter λ to two labelled datasets $\mathbf{Y}_{\mathcal{L}_t}$ and $\mathbf{Y}'_{\mathcal{L}_t}$ that differ by one label, associated with the node q_{t-1}^* , is bounded as:

$$|R_{|\mathbf{Y}_{\mathcal{L}_t}}^{+q} - R_{|\mathbf{Y}'_{\mathcal{L}_t}}^{+q}| \leq \frac{1}{|\mathcal{U}_t^q|} \sum_{i \in \mathcal{U}_t^q} \tilde{b}(\eta_q, \mathbf{w}_2, i).$$

Sketch of Proof. We define the random variable $\varphi_{i, \mathbf{Y}_{\mathcal{L}_t}}^{+q}$ which takes value $\varphi_{i, k, \mathbf{Y}_{\mathcal{L}_t}}^{+q}$ with probability $p(\mathbf{y}_q = k | \mathbf{Y}_{\mathcal{L}_t})$ for $k \in \{0, 1\}$. The difference in risk is:

$$\frac{1}{|\mathcal{U}_t^q|} \sum_{i \in \mathcal{U}_t^q} E_{\mathbf{y}_q} [\varphi_{i, \mathbf{Y}_{\mathcal{L}_t}}^{+q}] - E_{\mathbf{y}_q} [\varphi_{i, \mathbf{Y}'_{\mathcal{L}_t}}^{+q}], \quad (8)$$

where $E_{\mathbf{y}_q}$ denotes expectation over \mathbf{y}_q conditioned on the observed label set, either $\mathbf{Y}_{\mathcal{L}_t}$ or $\mathbf{Y}'_{\mathcal{L}_t}$. For query node q , for each $k_1, k_2 \in \{0, 1\}$, we learn weights \mathbf{w}_{1, k_1} and \mathbf{w}_{2, k_2} using $\{\mathbf{Y}_{\mathcal{L}_t} \cup \{\mathbf{y}_q = k_1\}\}$ and $\{\mathbf{Y}'_{\mathcal{L}_t} \cup \{\mathbf{y}_q = k_2\}\}$, respectively. For each $i \in \mathcal{U}_t^q$, we have:

$$\begin{aligned} |\varphi_{i, k_1, \mathbf{Y}_{\mathcal{L}_t}}^{+q} - \varphi_{i, k_2, \mathbf{Y}'_{\mathcal{L}_t}}^{+q}| &\leq |\sigma(\mathbf{w}_{1, k_1}^\top \tilde{\mathbf{x}}_i) - \sigma(\mathbf{w}_{2, k_2}^\top \tilde{\mathbf{x}}_i)|, \\ &\leq \tilde{b}(\eta_q, \mathbf{w}_2, i). \end{aligned} \quad (9)$$

Here the first inequality follows from the definition of $\varphi_{i, k_1, \mathbf{Y}_{\mathcal{L}_t}}^{+q}$ and the property that for $0 \leq p_1, p_2 \leq 1$, $|\min(p_1, 1 - p_1) - \min(p_2, 1 - p_2)| \leq |p_1 - p_2|$. The second inequality follows from Proposition (1), observing that the labels can differ only for nodes q_{t-1}^* and q .

Observe that for random variables X and Y taking values in $[a, b]$ and $[c, d]$, respectively, $|E_X[X] - E_Y[Y]| \leq \max(|a - d|, |b - c|)$. Applying this to (8) and then employing (9) leads to the stated bound on the risk error. \square

The following bound applies for the case of multiclass classification. For a given label set \mathbf{Y} , we learn weights $\mathbf{w}^{(k)}$ for each class $k \in \{1, \dots, K\}$ using L2-regularized binary one-vs-all logistic regression. The output prior to normalization for a given feature vector $\tilde{\mathbf{x}}_i$ is then $\sigma(\mathbf{w}^{(k)\top} \tilde{\mathbf{x}}_i)$. We then normalize by dividing by $C_i(\mathbf{w}) = \sum_{k=1}^K \sigma(\mathbf{w}^{(k)\top} \tilde{\mathbf{x}}_i)$ to obtain a probability vector. Let $\mathbf{w}_{2,k}^{(k')}$ be the weight vector learned for class k' using label data $\{\mathbf{Y}'_{\mathcal{L}_t} \cup \{\mathbf{y}_q = k'\}\}$. Let $\rho(\mathbf{w}_{2,k}, \eta, i) \triangleq \max_{k' \in K} \max(|b_\eta(\mathbf{w}_{2,k}^{(k')}, i)|, |b_{-\eta}(\mathbf{w}_{2,k}^{(k')}, i)|)$ for $b_{\pm\eta}(\mathbf{w}_2, i) \triangleq \sigma(\mathbf{w}_2^\top \tilde{\mathbf{x}}_i) - \sigma(\mathbf{w}_2^\top \tilde{\mathbf{x}}_i \pm 2\eta \|\tilde{\mathbf{x}}_i\|)$. Define

$$\tilde{\beta}(\mathbf{w}_2, \eta, i) \triangleq \max_{k,k'} \left(\left| \frac{\sigma(\mathbf{w}_{2,k}^{(k')\top} \tilde{\mathbf{x}}_i)}{C_i(\mathbf{w}_{2,k})} - \frac{\mathbf{w}_{2,k}^{(k')\top} \tilde{\mathbf{x}}_i - \rho(\mathbf{w}_{2,k}, \eta, i)}{C_i(\mathbf{w}_{2,k}) + 4\rho(\mathbf{w}_{2,k}, \eta, i)} \right|, \left| \frac{\sigma(\mathbf{w}_{2,k}^{(k')\top} \tilde{\mathbf{x}}_i)}{C_i(\mathbf{w}_{2,k})} - \frac{\mathbf{w}_{2,k}^{(k')\top} \tilde{\mathbf{x}}_i + \rho(\mathbf{w}_{2,k}, \eta, i)}{C_i(\mathbf{w}_{2,k}) - 4\rho(\mathbf{w}_{2,k}, \eta, i)} \right| \right).$$

Theorem 4.2. *The risk error arising from multiclass regression performed via repeated one-vs-all L2-regularized logistic regression with regularization parameter λ to labelled datasets $\mathbf{Y}_{\mathcal{L}_t}$ and $\mathbf{Y}'_{\mathcal{L}_t}$ that differ by one label, associated with the node q_{t-1}^* , is bounded as:*

$$|R_{|\mathbf{Y}_{\mathcal{L}_t}}^{+q} - R_{|\mathbf{Y}'_{\mathcal{L}_t}}^{+q}| \leq \frac{1}{|\mathcal{U}_t^{+q}|} \sum_{i \in \mathcal{U}_t^{+q}} \tilde{\beta}(\mathbf{w}_2, \eta_q, i), \quad (10)$$

for $\eta_q \triangleq \frac{1}{2\lambda} (\|\tilde{\mathbf{x}}_q\| + \|\tilde{\mathbf{x}}_{q_{t-1}^*}\|)$.

The proof is similar to that of Theorem 4.1, but more involved, and is provided in the supplementary material.

4.6. Combined method

The most extreme case of active learning is when we start with only one labelled node. In this scenario, the logistic regression model cannot make useful predictions until at least some nodes have been queried. To address this scenario, we combine our algorithm with a label-propagation method. The aim is to first use label-propagation when very few node labels are available, then switch to a combination of both algorithms when more information is available, and finally transition to the more accurate graph-cognizant logistic regression. Bayesian model averaging provides a mechanism to make this transition (Minka, 2002).

In Bayesian model averaging, we have V different classifiers and our belief is that one of these models is correct. We start with a prior $p(v)$ over each model. After observing data D , we compute the model evidence $p(D|v)$. Using Bayes' rule, we can compute the posterior $p(v|D) = p(v)p(D|v)/p(D)$ and then weight the predictions:

$$p(y_i|D) = \sum_{v=1}^V p(y_i, v|D) = \sum_{v=1}^V p(y_i|v, D)p(v|D).$$

In the context of expected error minimization active learning, we need to evaluate the risk associated with a query. We introduce a model-dependent zero-one risk, and in our combined method, we employ a model-averaged risk:

$$R_{|\mathbf{Y}_{\mathcal{L}}}^{+q} = \sum_{v=1}^V E_{\mathbf{y}_q} \left[R(\mathbf{Y}_{\mathcal{L}}, \mathbf{y}_q, v) \right] p(v|\mathbf{Y}_{\mathcal{L}}). \quad (11)$$

In order to compute (or approximate) this expression, we need to evaluate $p(v|\mathbf{Y}_{\mathcal{L}})$. Assuming that we have equal prior belief in the models available to us, this is equivalent to calculating the marginal likelihood $p(\mathbf{Y}_{\mathcal{L}}|v)$.

We incorporate two models, one based on label propagation and the other based on logistic regression. For the binary random field model that underpins the label propagation classifiers, there are no learnable model parameters (there is one fixed hyperparameter). Evaluating the evidence $p(\mathbf{Y}_{\mathcal{L}}|v)$ is thus equivalent to computing $p(\mathbf{Y}_{\mathcal{L}})$ under the BMRF model. This is a combinatorial problem, but we can factorize the joint probability into a chain rule of conditionals and use the same two-stage approximation (TSA) that is employed in (Jun & Nowak, 2016). Additional details are provided in the supplementary material. We denote this evidence approximation by $\lambda_{TSA, \mathbf{Y}_{\mathcal{L}}} \triangleq p(\mathbf{Y}_{\mathcal{L}}|BMRF)$.

For the logistic regression model, we are using $p(\mathbf{y}_i = k|v, \mathbf{W}) = \sigma(\mathbf{W}^\top \tilde{\mathbf{x}}_i)^{(k)}$. The joint probability of the complete labelled set is then evaluated as $p(\mathbf{Y}_{\mathcal{L}}|v, \mathbf{W}) = \prod_{\mathbf{y}_i \in \mathbf{Y}_{\mathcal{L}}} \sigma(\mathbf{W}^\top \tilde{\mathbf{x}})^{(k_i)}$, where k_i is the categorical index of \mathbf{y}_i . We then have $p(\mathbf{Y}_{\mathcal{L}}|v) = \int_{\Theta} \prod_{\mathbf{y}_i \in \mathbf{Y}_{\mathcal{L}}} \sigma(\mathbf{W}^\top \tilde{\mathbf{x}})^{(k_i)} p(\mathbf{W}) d\mathbf{W}$. To calculate the evidence, we thus need to integrate over the weight matrix \mathbf{W} , which is not analytically tractable. We choose to approximate $p(\mathbf{Y}_{\mathcal{L}}|v)$ by $p(\mathbf{Y}_{\mathcal{L}}|v, \mathbf{W}_{\mathbf{Y}_{\mathcal{L}}})$, and we denote this as $\lambda_{LG, \mathbf{Y}_{\mathcal{L}}} \triangleq \prod_{\mathbf{y}_i \in \mathbf{Y}_{\mathcal{L}}} \sigma(\mathbf{W}_{\mathbf{Y}_{\mathcal{L}}}^\top \tilde{\mathbf{x}})^{(k_i)}$. This leads to a sufficiently accurate approximation of the evidence for our purpose (which is just to achieve an adaptive balance between label propagation and SGC).

We use the TSA algorithm (Jun & Nowak, 2016) as the label propagation-based estimator, leading to a new combined approach for selecting the query node. We solve

$$\arg \min_{q \in \mathcal{U}_t} R_{|\mathbf{Y}_{\mathcal{L}}}^{+q} = \bar{\lambda}_{LG, \mathbf{Y}_{\mathcal{L}}} R_{|\mathbf{Y}_{\mathcal{L}}, LG}^{+q} + \bar{\lambda}_{TSA, \mathbf{Y}_{\mathcal{L}}} R_{|\mathbf{Y}_{\mathcal{L}}, TSA}^{+q},$$

where $\bar{\lambda}_{LG, \mathbf{Y}_{\mathcal{L}}}$ and $\bar{\lambda}_{TSA, \mathbf{Y}_{\mathcal{L}}}$ are our normalized estimates for the model evidences for the SGC model and the TSA label propagation model, respectively, after observing the data $\mathbf{Y}_{\mathcal{L}}$.

5. Experiments

We examine performance using five of the node classification benchmarks in (Shchur et al., 2018). Cora, Cite-seer (Sen et al., 2008) and Pubmed (Namata et al., 2012)

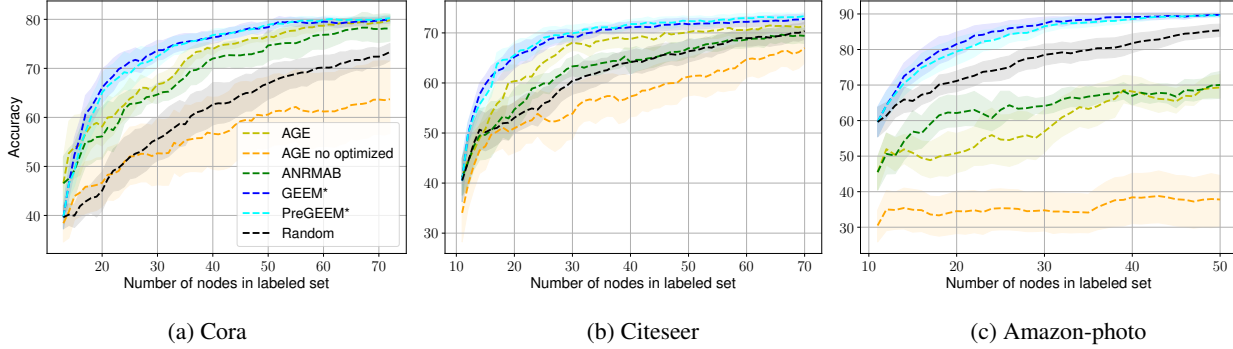


Figure 3. Comparison of performance of active learning algorithms for Experiment 1. Each point on a curve shows the mean classification accuracy achieved across 20 random partitions after the labelled set has been expanded to the indicated number of nodes. The shaded regions indicate 5/95 confidence intervals on the means derived using bootstrap.

are citation datasets. Nodes represent journal articles and an undirected edge is included when one article cites another. The node features are bag-of-words representations of article content. Amazon-Photo and Amazon-Computers are graphs based on customers’ co-purchase history records. For each dataset we isolate the largest connected component in the graph following (Shchur et al., 2018). The description of the dataset statistics is shown in Table 1.

Table 1. Statistics of evaluation datasets.

Dataset	Classes	Features	Nodes	Edges	Edge Density
Cora	7	1,433	2,485	5,069	0.04%
Citeseer	6	3,703	2,110	3,668	0.04%
Pubmed	3	500	19,717	44,324	0.01%
Am-comp.	10	767	13,381	245,778	0.07%
Am-photo	8	745	7,487	119,043	0.11%
Microwave	2	19	322	5,753	5.54%

5.1. Baselines

We compare the following active learning algorithms: (i) **Random**: This baseline chooses a node to query by uniform random selection, and then performs classification using SGC; (ii) **AGE**: The graph neural network based algorithm proposed by (Cai et al., 2017); (iii) **ANRMAB**: The graph neural network algorithm proposed by (Gao et al., 2018a), in which a multi-arm bandit is used to adapt the weights assigned to the different metrics used when constructing the score to choose a query node; (iv) **TSA**: The label-propagation algorithm based on a two-stage approximation of the BMRF model (Jun & Nowak, 2016); (v) **EC-TV, EC-MSD**: The label-propagation algorithms based on a Gaussian random field approximation to the BMRF model (Berberidis & Giannakis, 2018); (vi) **GEEM**: The proposed algorithm based on SGC and expected error min-

imization; (vii) **PreGEEM**: The proposed algorithm with preemptive queries; (viii) **Combined**: The proposed combined algorithm that uses Bayesian model averaging to adaptively merge SGC and label propagation in an EEM framework.

5.2. Experimental Settings

For each experiment, we report the average over 20 trials with different random partitions. All GCNs and SGCs have 2 layers. The weight-adapting parameter of AGE is set to the values in (Cai et al., 2017) and to 0.995 for non-included datasets. For the larger datasets, Am-Photo, Am-Comp and Pubmed, we reduce computational complexity for GEEM and PreGEEM by evaluating risk using a subset of 500 nodes, selected randomly in an approach similar to (Roy & McCallum, 2001). This has minimal impact on performance. The GCN hyperparameters are set to the values found by (Shchur et al., 2018) to be the best performing hyperparameter configurations. Early stopping is not employed because access to a validation set is not a reasonable assumption in an active learning setting. We also include a “non-optimized” version of AGE; this is because, in practice, we would usually not have access to the tuned hyperparameters provided by (Shchur et al., 2018), because these are derived using a large validation set. For the non-optimized version of AGE, the hyperparameter configuration for each trial was randomly selected from the values considered in the grid search of (Shchur et al., 2018).

Experiment 1: Initial Labelled Set, Transductive: Each algorithm is initially provided with a small set of randomly chosen labelled nodes. We evaluate performance on a set of test nodes comprising 20% of the unlabelled set. The algorithms cannot query nodes from this evaluation set. Algorithms have access to the entire topology and all node features. For the Cora and Citeseer datasets, we start with

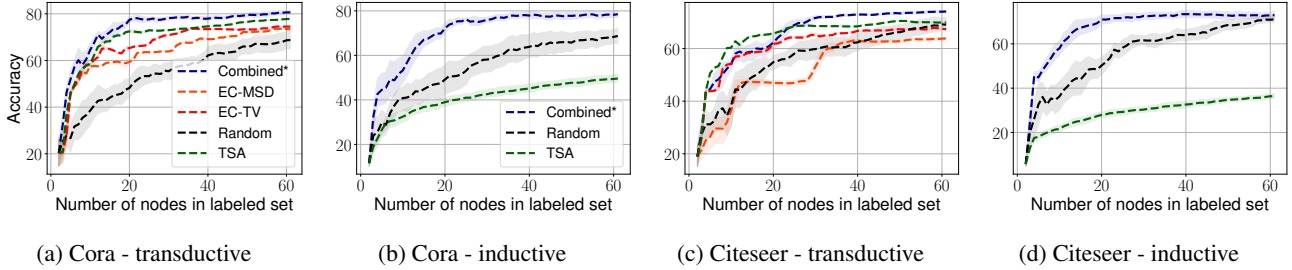


Figure 4. Performance comparison between the label propagation algorithms and the proposed combined model-averaging expected error minimization method for the case when the initial label set consists of one random node. In the transductive setting, accuracy is evaluated across all unlabelled nodes; in the inductive setting, accuracy is evaluated on a held-out test set of nodes.

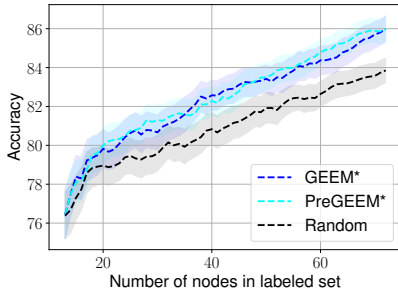


Figure 5. Performance of the active learning algorithms for detection of faulty links in the microwave link network.

0.5% of labelled nodes. We reduce this to 0.01% for the larger datasets to achieve similar initial set sizes.

Experiment 2: Single Labelled Node, Transductive and Inductive: Algorithms start with a single random labeled node. We examine two settings. In the transductive setting, algorithms know the entire graph and can access all features. Performance is assessed over all unlabelled nodes. In the inductive setting, a portion of the graph is held out for testing; the algorithms do not have access to the features and topology information for these nodes.

5.3. Results and Discussion

Experiment 1: Figure 3 and Table 2 show how the algorithms’ accuracies change as more nodes are added to an initial labelled set of size 10-20 nodes. Label propagation algorithm performances are not shown because they are outperformed by the GCN methods for this scenario. For all presented datasets, the proposed algorithms outperform the other GCN-based methods. This holds even for the cases when the hyperparameters of the GCN have been optimized using a validation set. When the hyperparameters are not tuned, the performance of the AGE algorithm deteriorates dramatically. It is better to choose the query node randomly. AGE outperforms the ANRMAB algorithm for the datasets

where its weight-adapting parameter was tuned (Cora, Citeseer and Pubmed). The Random baseline and the proposed GEEM method use the same classifier, so they differ only in the nodes that are queried. Choosing an informative set of nodes using our proposed methods leads to a substantially improved accuracy in all cases. For the Cora dataset, the optimized GCN classifier initially outperforms the SGC model. However GEEM quickly outperforms as more nodes are queried, showing that the selection algorithm is more effective. Comparing PreGEEM performance against GEEM, it is clear that the approximation has very little impact; there is no clear performance difference between the two.

Experiment 2: Figure 4 compares the performance of the proposed Combined method with the label propagation algorithms. In the transductive setting, the proposed method is much better than Random selection. Since it incorporates the TSA technique, its performance is similar to TSA when few nodes have been queried. As the number of labels increases, there starts to be a small but significant improvement in accuracy. The inability of the label propagation methods to adapt to the inductive setting is shown clearly in Figures 4(b) and 4(d). In order to choose effective nodes to query, these methods need to know the topology of the entire graph. By contrast, the Combined method, which incorporates graph-based logistic regression, achieves similar performance in both inductive and transductive settings.

5.4. Practical Application

To give a concrete motivating application for PreGEEM, we also report the results of experiments on a private company dataset obtained from measurements of a microwave link network. Currently, faulty links are identified by human operators who must process lengthy performance log files. The identification or labelling of a faulty link takes a few minutes. Link performances vary substantially over time, so it is necessary to repeatedly label data. It is desirable to automate the faulty link detection procedure by training a classifier. Active learning has the potential to substantially

Table 2. Experiment 1 and Practical Application: Average accuracy at different budgets. Asterisks indicate that a Wilcoxon ranking test showed a significant difference (at 5% significance level) between the marked method and the best performing baseline.

budget b	0	1	10	30	b
Cora					(60)
GEEM*	39.6	46.5	69.8*	77.2*	79.9
PreGEEM*	39.6	46.5	68.2*	77.1*	80.3
Random	39.6	40.2	49.7	63.0	73.3
AGE	46.6	52.7	61.6	74.9	79.8
ANRMAB	46.6	47.5	59.1	72.7	78.1
Citeseer					(60)
GEEM*	40.5	49.7*	65.8*	71.2	72.8
PreGEEM*	40.5	49.7*	66.5*	71.8	73.3
Random	40.5	44.1	53.8	64.4	70.4
AGE	41.2	44.7	60.5	69.1	71.4
ANRMAB	41.2	44.1	55.7	64.6	69.4
Pubmed					(40)
GEEM*	52.3	58.1	72.6	77.6	78.7
PreGEEM*	52.3	58.1	69.3	77.2	78.0
Random	52.3	54.1	64.7	72.3	73.9
AGE	57.3	60.8	70.4	76.7	78.1
ANRMAB	57.3	58.8	69.5	74.1	75.7
Am-photo					(40)
GEEM*	59.6*	64.3*	82.4*	89.2*	90.7*
PreGEEM*	59.6*	64.3*	80.3*	88.8*	89.6
Random	59.6*	61.4*	72.0	82.3	87.6
AGE	45.5	52.0	51.5	67.8	69.3
ANRMAB	45.5	50.6	62.6	67.8	70.0
Am-comp.					(40)
GEEM*	54.6*	59.8*	68.8*	74.8*	76.8*
PreGEEM*	54.6*	59.8*	68.4*	76.5*	77.5*
Random	54.6*	57.7*	65.9	72.8	73.3
AGE	47.1	41.5	51.6	52.4	53.3
ANRMAB	47.1	49.4	54.6	58.7	58.5
Microwave					(60)
GEEM*	76.4	77.5	80.1	82.9*	86.0*
PreGEEM*	76.4	77.5	80.3	82.4*	86.0*
Random	76.4	76.7	79.1	81.0	83.9
AGE	69.1	68.3	70.3	70.3	75.1
ANRMAB	69.1	67.2	72.3	73.5	73.2

reduce the time an operator must devote to the labelling task each week. For graphs the size of common microwave link networks, the GEEM algorithm can return a query in approximately one to two minutes, so this is an example where the PreGEEM algorithm can compute the next query during the labelling process.

The graph is constructed directly from the physical topology and is important because graph-based classification significantly outperforms classification algorithms that ignore the network. The features are link characteristics such as received signal strength and signal distortion metrics. Table 1 provides the statistics of the dataset. We consider an

experiment where an initial labelled set of 8 links is provided, and the active learning algorithm must identify query nodes. Table 2 and Figure 5 compare the performance of GEEM, PreGEEM, Random, AGE and ANRMAB. AGE and ANRMAB performs much worse than random selection because the GCN is inaccurate for a small number of labels. GEEM and PreGEEM achieve a small but significant improvement.

6. Conclusion

We have introduced an active learning algorithm for node classification on attributed graphs that uses SGC (a linearized GCN) in an expected error minimization framework. Numerical experiments demonstrate that the proposed method significantly outperforms existing active learning algorithms for attributed graphs without relying on a validation set. We also proposed a preemptive algorithm that can generate a query while the oracle is labelling the previous query, and showed experimentally that this approximation does not impact the performance.

References

- Berberidis, D. and Giannakis, G. B. Data-adaptive active sampling for efficient graph-cognizant classification. *IEEE Trans. Signal Processing*, 66:5167–5179, Oct. 2018.
- Cai, H., Zheng, V. W., and Chang, K. C. Active learning for graph embedding. *arXiv preprint arXiv:1705.05085*, 2017.
- Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proc. Advances in Neural Information Processing Systems*, pp. 3844–3852, Barcelona, Spain, Dec. 2016.
- Gal, Y., Islam, R., and Ghahramani, Z. Deep bayesian active learning with image data. In *Proc. Int. Conf. on Machine Learning*, pp. 1183–1192, Sydney, Australia, Aug. 2017.
- Gao, H., Wang, Z., and Ji, S. Large-scale learnable graph convolutional networks. In *Proc. Int. Conf. on Knowledge Discovery & Data Mining*, pp. 1416–1424, London, United Kingdom, Aug. 2018a.
- Gao, L., Yang, H., Zhou, C., Wu, J., Pan, S., and Hu, Y. Active discriminative network representation learning. In *Proc. Int. Joint Conf. Artificial Intelligence*, pp. 2142–2148, Stockholm, Sweden, Jul. 2018b.
- Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Proc. Advances in Neural Information Processing Systems*, pp. 1024–1034, Long Beach, CA, US, Dec. 2017.

- Hoi, S. C. H., Jin, R., Zhu, J., and Lyu, M. R. Batch mode active learning and its application to medical image classification. In *Proc. Int. Conf. on Machine Learning*, pp. 417–424, Pittsburgh, PA, US, Jun. 2006.
- Ji, M. and Han, J. A variance minimization criterion to active learning on graphs. In *Proc. Int. Conf. Artificial Intelligence and Statistics*, pp. 556–564, La Palma, Canary Islands, Apr. 2012.
- Jun, K. and Nowak, R. Graph-based active learning: A new look at expected error minimization. In *Proc. IEEE Global Conf. Signal and Information Proc.*, pp. 1325–1329, Dec. 2016.
- Kipf, T. and Welling, M. Semi-supervised classification with graph convolutional networks. In *Proc. Int. Conf. Learning Representations*, Toulon, France, Apr. 2017.
- Kurzendorfer, T., Fischer, P., Mirshahzadeh, N., Pohl, T., Brost, A., Steidl, S., and Maier, A. Rapid interactive and intuitive segmentation of 3d medical images using radial basis function interpolation. *Journal of Imaging*, 3:56, Nov. 2017.
- Liu, Z., Chen, C., Li, L., Zhou, J., Li, X., and Song, L. Geniepath: Graph neural networks with adaptive receptive paths. In *Proc. AAAI Conf. on Artificial Intelligence*, pp. 4424–4431, Honolulu, HI, US, Jan. 2019.
- Ma, Y., Garnett, R., and Schneider, J. σ -optimality for active learning on Gaussian random fields. In *Proc. Adv. Neural Inf. Proc. Systems*, pp. 2751–2759, Lake Tahoe, NV, US, Dec. 2013.
- Minka, T. Bayesian model averaging is not model combination. MIT Media Lab Note, 2002.
- Namata, G., London, B., Getoor, L., and Huang, B. Query-driven active surveying for collective classification. In *Proc. Workshop on Mining and Learning with Graphs, Int. Conf. Machine Learning*, 2012.
- Parisot, S., Ktena, S. I., Ferrante, E., Lee, M., Guerrero, R., Glocker, B., and Rueckert, D. Disease prediction using graph convolutional networks: Application to autism spectrum disorder and Alzheimer’s disease. *Medical Image Analysis*, 48:117–130, Aug. 2018.
- Roy, N. and McCallum, A. Toward optimal active learning through sampling estimation of error reduction. In *Proc. Int. Conf. on Machine Learning*, pp. 441–448, San Francisco, CA, USA, June 2001.
- Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. Collective classification in network data. *AI Magazine*, 29(3):93, Sep. 2008.
- Settles, B. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.
- Shchur, O., Mumme, M., Bojchevski, A., and Günnemann, S. Pitfalls of graph neural network evaluation. In *Relational Representation Learning Workshop, NeurIPS 2018*, Montréal, Canada, Dec. 2018.
- Sivan, H., Gabel, M., and Schuster, A. Online linear models for edge computing. In *Proc. Eur. Conf. Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, 2019.
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *Proc. Int. Conf. Learning Representations*, Vancouver, Canada, Apr. 2018.
- Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., and Weinberger, K. Simplifying graph convolutional networks. In *Proc. Int. Conf. Machine Learning*, pp. 6861–6871, Long Beach, CA, US, Jun. 2019.
- Zhu, X., Ghahramani, Z., and Lafferty, J. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. Int. Conf. on Machine Learning*, pp. 912–919, Washington, DC, USA, Aug. 2003a.
- Zhu, X., Lafferty, J., and Ghahramani, Z. Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions. In *Proc. Workshop on The Continuum from Labeled to Unlabeled Data (ICML)*, pp. 58–65, Washington, DC, US, Aug. 2003b.
- Zhuang, C. and Ma, Q. Dual graph convolutional networks for graph-based semi-supervised classification. In *Proc. Int. World Wide Web Conf.*, pp. 499–508, Lyon, France, Apr. 2018.