Proceedings of Machine Learning Research 129:561–576, 2020 ACML 2020

MetAL: Active Semi-Supervised Learning on Graphs via Meta-Learning

Kaushalya Madhawa KAUSHALYA@NET.C.TITECH.AC.JP and
Tsuyoshi Murata MURATA@C.TITECH.AC.JP
Department of Computer Science, Tokyo Institute of Technology, Tokyo 152-8552, Japan

Editors: Sinno Jialin Pan and Masashi Sugiyama

Abstract

The objective of active learning (AL) is to train classification models with less labeled instances by selecting only the most informative instances for labeling. The AL algorithms designed for other data types such as images and text do not perform well on graph-structured data. Although a few heuristics-based AL algorithms have been proposed for graphs, a principled approach is lacking. In this paper, we propose MetAL, an AL approach that selects unlabeled instances that directly improve the future performance of a classification model. For a semi-supervised learning problem, we formulate the AL task as a bilevel optimization problem. Based on recent work in meta-learning, we use the meta-gradients to approximate the impact of retraining the model with any unlabeled instance on the model performance. Using multiple graph datasets belonging to different domains, we demonstrate that MetAL efficiently outperforms existing state-of-the-art AL algorithms.

Keywords: Active Learning, Graph Neural Networks

1. Introduction

The performance of a classification model depends on the quality and quantity of training data, often requiring a huge labeling effort. With ever-increasing amounts of data, active learning (AL) is gaining the attention of researchers as well as practitioners as a way to reduce the effort spent on labeling data instances. An AL algorithm selects a set of unlabeled instances based on an informative metric, gets their labels, and updates the labeled dataset. Then the classification model is retrained using the acquired labels. This process is repeated until a desirable level of performance (e.g. accuracy) is reached.

In this paper, we consider the task of applying AL for semi-supervised problems. In a semi-supervised learning problem the learning algorithm can utilize all data instances including the unlabeled ones. Only the labels of unlabeled instances are not known. We evaluate our approach on classifying nodes on attributed graphs. Reducing the number of labeled nodes required in node classification can benefit a variety of practical applications such as in recommender systems (Ying et al., 2018; Rubens et al., 2015) and text classification (Yao et al., 2019).

An *acquisition function* is used to evaluate the informativeness of an unlabeled instance. Since quantifying the *informativeness* of an instance is not straightforward, a multitude of heuristics have been proposed in AL literature (Settles, 2009). For example, *uncertainty sampling* selects instances which the model is most uncertain about (Houlsby et al., 2011).

The most common method is to select the unlabeled instance corresponding to the maximum entropy over the class probabilities predicted by the model. However, such heuristics are not flexible to adapt to the distribution of data and can not exploit inherent characteristics of a given dataset. Often, the performance of heuristic active learners is not consistent across different datasets, sometimes worse than random selection of unlabeled instances.

Compared to applications of AL on image data, only a limited number of AL models have been developed for graph data. Previous work on applying AL on graph data (Gu and Han, 2012; Bilgic et al., 2010; Ji and Han, 2012) depend on earlier classification models such as Gaussian random fields, in which the features of nodes are not being used. Therefore, selecting query nodes uniformly in random coupled with a recent graph neural network (GNN) model can easily outperform such AL models. AL models that use recent GNN architectures (Cai et al., 2017; Gao et al., 2018) are limited and they rely on linear combinations of uncertainty and various heuristics such as node centrality measures.

We overcome this issue by directly incorporating the performance of the classifier into the acquisition function in semi-supervised learning problems. We build our work motivated by the framework of *expected error reduction* (EER) (Roy and McCallum, 2001; Guo and Schuurmans, 2008; Mac Aodha et al., 2014), in which the objective is to query instances which would maximizes the expected performance gain. Original EER formulation is extremely time consuming and is not practical to be used with neural network classifiers. We formulate this objective as a bilevel optimization problem (Colson et al., 2007; Franceschi et al., 2018). Based on recent advances in meta-learning (Finn et al., 2017), we utilize meta-gradients to make this optimization efficient. Zügner and Günnemann (2019) propose using meta-gradients for modeling an adversarial attack on GNNs. Our motivation in using meta-gradients is the opposite, evaluating the importance of labeling each unlabeled instance. In section 4, with empirical evidence, we show that MetAL[1], our proposed algorithm significantly outperforms existing AL algorithms.

Our contributions are:

1. We introduce MetAL, a novel active learning algorithm based on the expected error reduction principle.
2. We discuss the importance of performing exploration in AL and introduce a simple count-based exploration term.
3. We demonstrate that our proposed algorithm MetAL can consistently outperform state-of-the-art AL algorithms on a variety of real world graphs.

2. Our Framework

2.1. Problem Setting

In this paper, we apply AL for the multi-class node classification of a given undirected attributed graph G of N nodes. The graph G consists of an adjacency matrix $A \in \{0,1\}^{N \times N}$ and a node attribute matrix $X \in \mathbb{R}^{N \times F}$, where F is the number of attributes. Labels of a small set of nodes $\mathcal{L}$ are given initially and labels of rest of the nodes $\mathcal{U}$ are unknown.

1. The code is available at https://github.com/Kaushalya/metal

A labeled node is assigned a label in $\{1, 2, \ldots C\}$, where C is the number of classes. The objective of a learner is to learn a function $f_\theta(x_i)$ which predicts the class label of a given test node $i \in \mathcal{U}$. This f_θ function can be any node classification algorithm. Graph neural networks (Kipf and Welling, 2017; Wu et al., 2019) (GNN) are commonly used in the present day. Parameters θ of the model are estimated by minimizing a loss function, usually using a gradient-based optimization algorithm.

We consider a *pool-based active learning* setting, in which the labeled dataset $\mathcal{L}$ is much smaller compared to a large pool of unlabeled items $\mathcal{U}$. We can acquire the label of any unlabeled item by querying an oracle (e.g. a human annotator) at a uniform cost per item. Suppose we are given a query budget K, such that we are allowed to query labels of a maximum of K unlabeled items. An optimal active learner selects the set of K items which maximizes the expected performance gain of the classification model upon retraining it with their labels. Selection of K items for querying is done in an iterative manner such that in each iteration an instance q is queried and the model is retrained with its label.

2.2. Optimization Problem

We define our objective as finding q unlabeled items which maximizes the likelihood of labeled instances while minimizing the uncertainty of label predictions of the unlabeled instances $\mathcal{U} \setminus q$. For any $q \in \mathcal{U}$ we estimate this objective of the model after training it on q. Training on an item (x_q, y_q) updates model parameters $\hat{\theta}$ to $\hat{\theta}^{+(x_q, y_q)}$ such that

$$\hat{\theta}^{+(x_q,y_q)} = \arg\min_\theta l(f_\theta(G), Y_\mathcal{L} \cup y_q), \tag{1}$$

where l is the loss function (e.g. cross-entropy). We can write our objective as an optimization problem:

$$q^* = \arg\min_q \mathcal{E}(f_{\hat{\theta}^{+(x_q,y_q)}}), \tag{2}$$

where $\mathcal{E}$ is a cost function defined as

$$\mathcal{E}(f_{\hat{\theta}^{+(x_q,y_q)}}) = l(f_{\hat{\theta}^{+(x_q,y_q)}}(G), Y_{\mathcal{L}\cup q}) + \mathbf{H}([f_{\hat{\theta}^{+(x_q,y_q)}}(G)]_{\mathcal{U}\setminus q}), \tag{3}$$

in which we minimize the loss over labeled instances combined with $\mathbf{H}([f_{\hat{\theta}^{+(x_q,y_q)}}(G)]_{\mathcal{U}\setminus q})$, the entropy of unlabeled instances.

Since the label y_q of an unlabeled instance q is unknown, we compute the expected loss over all possible labels. We rewrite Equation (3) as

$$\arg\min_q \sum_{k=1}^{C} P(\hat{y}_q = k | G, Y_\mathcal{U}))\mathcal{E}(f_{\hat{\theta}^{+(x_q,y_q=k)}}). \tag{4}$$

In this case, we select the instance x_q which minimizes the expected value of $\mathcal{E}$. $\hat{\theta}^{+(x_q,y_q=k)}$ denotes the parameters of a model trained with instance q having the label k.

2.3. Meta-learning Approach

Since the label of an item $q \in \mathcal{U}$ is unknown, we use the posterior class probabilities $\hat{y}_q$ as a proxy for y_q. This approach requires training a separate model for each possible label of each unlabeled item ($N_\mathcal{U} \times C$). Training this many models is prohibitively time consuming.

As a solution, we let the labels to be real-valued ($Y_{\mathcal{U}} \in [0,1]^{N_{\mathcal{U}} \times C}$) and we estimate the impact of a query q having the label k ($y_q = k$) by training a model with label $\hat{y}_{q,k}$ upweighted by a small perturbation δ such that $(x_q, y_q = \hat{y}_q + \hat{y}_q \cdot \delta_{q,k})$, where $\delta_{q,k} \in \mathbb{R}$ is the perturbation added to label k. This idea is motivated by the use of perturbations in the feature space for finding training instances responsible for a given prediction (Koh and Liang, 2017). In contrast, our objective is to find unlabeled instances which incur the greatest impact on the performance on test instances, once their labels are known. We re-purpose the use of perturbations to understand the impact an unlabeled instance q may have on the model performance if it has the label y_q. We rewrite Equation (1) as

$$\hat{\theta}^{+(x_q, y_q=k)} = \arg\min_{\theta} l(f_\theta(G), Y_{\mathcal{L}} \cup \hat{Y}_q \odot (1 + \delta_{q,k})). \tag{5}$$

We quantify the impact of retraining the model with (x_q, y_q) added to the labeled set as the change in loss

$$\Delta\mathcal{E}_{q,k} = \mathcal{E}(f_{\hat{\theta}^{+(x_q, y_q=k)}}) - \mathcal{E}(f_{\hat{\theta}}), \tag{6}$$

and the expected change of loss for querying the item q by

$$\Delta\mathcal{E}_q = \sum_{k=1}^{C} P(\hat{y}_q = k | G, Y_{\mathcal{L}}) \ \Delta\mathcal{E}_{q,k}. \tag{7}$$

$P(y_q = k|G, Y_{\mathcal{L}})$ is the posterior class probabilities of the current model $f_{\hat{\theta}}$ and it is estimated with

$$P(y_q = k|G, Y_{\mathcal{L}}) = \text{softmax}(f_{\hat{\theta}}(G)). \tag{8}$$

When $\delta_{q,k}$ is arbitrarily small, this change can be computed as the gradient of the loss with respect to the label perturbation $\delta_{q,k}$, $\Delta\mathcal{E}_{q,k} \rightarrow \nabla_{\delta_{q,k}} \mathcal{E}(f_{\hat{\theta}_{q,k}}, Y_{\mathcal{U}\setminus q})$. We rewrite Equation (7) using gradient as

$$\Delta\mathcal{E}_q = \sum_{k=1}^{C} P(\hat{y}_q = k | G, Y_{\mathcal{L}}) \ \nabla_{\delta_{q,k}} \mathcal{E}(f_{\hat{\theta}^{+(x_q, y_q=k)}}). \tag{9}$$

The term $\Delta\mathcal{E}_q$ quantifies the impact of labeling a query q. This simplifies the active learning problem to finding the item corresponding to the minimum expected meta-gradient $\Delta\mathcal{E}_q$ (Equation (9)) such that

$$q^* = \arg\min_{q} \Delta\mathcal{E}_q. \tag{10}$$

Here, a negative valued expected meta-gradient corresponds to a model with lower expected loss. In other words, we need to find a query q which maximizes the negative of the expected gradient ($-\Delta\mathcal{E}_q$).

Equation (5) and Equation(10) form a bilevel optimization problem (Colson et al., 2007). Calculating the meta-gradients as in Equation (9) involves calculation of two gradients in a nested order, the inner one for optimizing the model parameters $\hat{\theta}_q$ for perturbed labels and the outer one for calculating the gradient with respect to the perturbation $\delta_{q,k}$. Therefore, the expected value of $\mathcal{E}$ indirectly depends on δ via $\hat{\theta}^{+(x_q, y_q=k)}$. This is similar to the computation of meta-gradients in meta-learning approaches used for few-shot learning (Finn et al., 2017) or hyper-gradients in gradient-based hyper-parameter optimization (Franceschi

et al., 2018). It should be noted that, unlike in few-shot learning, we calculate meta-gradients with respect to a perturbation added to the labels instead of differentiating with respect to model parameters.

Calculating $\Delta\mathcal{E}_q$ for each unlabeled node with Equation(9) is inefficient for practical applications of this algorithm.We address this problem by selecting a subset of unlabeled items having higher prediction uncertainty to estimate the model uncertainty in Equation (3) and remaining unlabeled items as query items $\mathcal{Q}$. We add a small perturbation $\delta_{\mathcal{Q}} \in \mathbb{R}^{N_{\mathcal{Q}} \times C}$ to the labels of $\mathcal{Q}$ items and retrain the model with these perturbed labels. With vector notation we can rewrite Equation (5) as

$$\hat{\theta}^{+(x_{\mathcal{Q}}, \hat{Y}_{\mathcal{Q}} \odot (1+\delta))} = \arg\min_{\theta} l(f_\theta(G), Y_{\mathcal{L}} \cup \hat{Y}_{\mathcal{Q}} \odot (1+\delta)). \tag{11}$$

Then we calculate the cost $\mathcal{E}$ and its gradient with respect to $\delta_{\mathcal{Q}}$ ($\nabla_{\delta_{\mathcal{Q}}}$). $\nabla_{\delta_{\mathcal{Q}}}$ is a real valued matrix, in which a row q corresponds to an unlabeled instance $q \in \mathcal{Q}$ and a column k corresponds to a label $k \in 1, \ldots, C$. For example, the gradient vector of query instance q belonging to class k can be expressed as $\nabla_{\delta_{q,k}} = [\nabla_{\delta_{\mathcal{Q}}}]_{[q,k]}$. We use the notation $[\nabla_{\delta_{\mathcal{Q}}}]_{[q,k]}$ to denote the element at q^{th} row and k^{th} column.

In our experiments, we use the top 10% unlabeled items with the largest prediction entropy to estimate the model entropy and the rest of unlabeled items as $\mathcal{Q}$. Our algorithm is shown in Algorithm 1. We select the node corresponding to the minimal meta-gradient and retrieve its label from the oracle. We add this node and its label to the labeled set and retrain the model.

2.4. The Importance of Exploration

After each acquisition step, the classifier is trained on a limited number of labeled instances, which in turn are selected by the active learner. Hence, the selected labeled instances tend to bias towards instances evaluated to be 'informative' by the active learner. In MetAL, the active learner selects the instance which minimizes the meta-gradient. Therefore, the distribution of labeled instances is far from the true underlying distribution. The active learner cannot observe the consequences of selecting an instance which has lower 'informativeness'. Therefore, it is desirable to query a few instances in addition to the ones maximizing our selection criteria. This step is known as 'exploration' while selecting the instance maximizing the criteria is 'exploitation'. Intuitively, an active learner should perform more exploration initially, so it can have a better view of the true distribution of data.

This problem is known as *exploration vs exploitation tradeoff* in sequential decision-making problems. Solving this tradeoff requires the learner to acquire potentially sub-optimal instances (i.e., exploration) in addition to the optimal ones. This problem is studied under the framework of multi-armed bandits problem (Lattimore and Szepesvári, 2020) (MAB). A multitude of approaches is used in solving online learning problems modeled as MAB problems. ϵ-greedy, upper confidence bounds (UCB) (Auer, 2002), and Thompson sampling (Thompson, 1933) are few of the frequently used techniques. Influenced by count-based approaches proposed for MAB problems, we introduce a simple exploration term in addition to the exploitation performed using the meta-gradients. We define the exploration term of

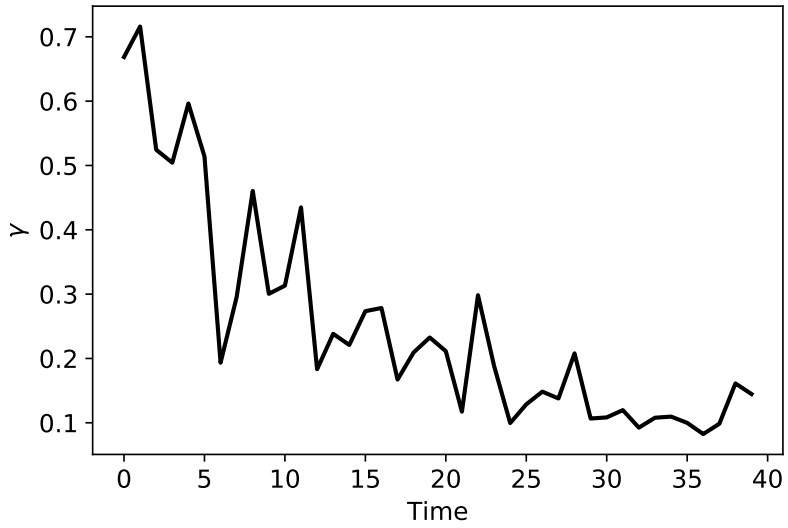


Figure 1: Variation of γ_t over time. Average of 10 random samples of Beta(α, β_t) distribution. We set the value of α to 1 and increase β_t with time t.

an instance i as the logarithm of the number of unlabeled neighboring nodes of i. This term encourages the learner to sample nodes from less labeled neighborhoods. Since this term and the gradient calculated in Equation (9) are on different scales, we normalize both of these quantities into $[0, 1]$ range and get $\phi_{exp}(i)$ and $\phi_{\nabla}(i)$ respectively. We linearly combine these normalized quantities to get the criterion for acquiring nodes as

$$\phi(i) = (1 - \gamma_t) \cdot \phi_{\nabla}(i) + \gamma_t \cdot \phi_{exp}(i), \tag{12}$$

where the exploration coefficient γ_t is a hyper-parameter that balances exploration and exploitation. Setting γ_t to 0 corresponds to pure exploration disregarding the feedback of the classifier (i.e. meta-gradient information). On the other hand, $\gamma_t = 1$ is equivalent to pure exploitation selecting the node with the minimum meta-gradient. We vary the value of γ_t with time, such that more exploration is performed during the initial acquisition steps followed by more exploitation in later rounds. To achieve this effect, we assume γ_t is sampled from a Beta distribution such that $\gamma_t \sim Beta(\alpha, \beta_t)$. We linearly increase the value of β_t over iterations of acquisitions to achieve the required effect of γ_t. As shown in Zhang et al. (2017), we observe smoother performance compared to setting the value of γ_t deterministically. Figure 1 shows how the value of γ_t varies over time in average.

3. Experiments

3.1. Data

We evaluate our proposed approach on 6 datasets belonging to different domains. CiteSeer, PubMed, and CORA (Sen et al., 2008) are commonly used citation graphs. Each of these graphs is made of documents as nodes and citations as edges between them. If one document

Algorithm 1 MetAL: Meta Learning Active Node Classification.

Input: Graph $G = (A, X)$, Query budget K, Initial labels $Y_{\mathcal{L}}$
Output: An improved model
$\theta \leftarrow$ train model on G with known labels $Y_{\mathcal{L}}$
for $i \leftarrow 1$ to $n_q = K$ **do**
  Calculate posterior class probabilities $\hat{Y}$ with the current model
  Sample a set of $N_{\mathcal{Q}}$ instances $\mathcal{Q}$ from $\mathcal{U}$
  Train a model with perturbed labels of $\mathcal{Q}$ instances with Equation (11)
  Calculate meta-gradient $\nabla_{\delta_{\mathcal{Q}}}$
  Select the best instance q^* using Equation (12)
  Query the oracle and retrieve the label Y_{q^*}
  Update label set $Y_{\mathcal{L}} \leftarrow Y_{\mathcal{L}} \cup Y_{q^*}$
  Retrain the model $\theta \leftarrow \arg\min_\theta l(f_\theta(G), Y_{\mathcal{L}})$
end for
Return θ

Table 1: Dataset statistics. Labeling rate as a percentage of total nodes is shown within brackets.

Dataset	Nodes	Classes	Features	Labels (%)
CiteSeer	2110	6	3703	12 (0.56)
PubMed	19717	3	500	6 (0.03)
CORA	2485	7	1433	14 (0.56)
Amazon Computers	13752	10	767	20 (0.14)
Co-author Physics	34493	5	8415	10 (0.03)
Co-author CS	18333	15	6805	30 (0.16)

cites another, they are linked by an edge. Each node contains bag-of-words features of its text as its attributes. Co-author CS and Co-author Physics are co-authorship graphs constructed from Microsoft Academic Graph. Nodes are authors, two authors are linked by an edge if they have co-authored a paper. Node features correspond to the keywords of the papers authored by a particular author. An author's most active field of study is used as the node label. Amazon Computers is a subgraph of the Amazon co-purchase graph (McAuley et al., 2015). Products are represented as nodes, two nodes are connected by an edge of those two products are frequently bought together. Node features correspond to product reviews encoded as bag-of-words. The product category is the node label.

For each dataset, we randomly select two nodes belonging to each label as the initial labeled set $V_{\mathcal{L}}$. We leave 5% of the rest of the unlabeled nodes as the test set. The remaining unlabeled nodes $V_{\mathcal{U}}$ qualify to be queried. The size of the initial labeled set and its size as a fraction of the total nodes (labeling rate) are shown in Table 1.

3.2. Model

We evaluate the effectiveness of MetAL, the proposed algorithm using a two-layer GCN model (Kipf and Welling, 2017) with 64 hidden units and SGC (Wu et al., 2019), a simplified

GNN architecture that does not include a hidden layer and nonlinear activation functions. In all experiments, we use the default hyper-parameters used in GNN literature (e.g. learning rate = 0.01). We do not perform any dataset-specific hyper-parameter tuning since hyper-parameter tuning while training a model with AL can lead to label inefficiency (Ash et al., 2020). We use the following algorithms in our comparison:

- **Random:** Selects an unlabeled node randomly.
- **PageRank:** Selects the unlabeled node with the largest PageRank centrality value.
- **Degree**: Selects the unlabeled node with the largest degree centrality value.
- **Entropy:** Calculates the entropy of predictions of the current model over unlabeled nodes and select the node corresponding to the largest entropy value.
- **AGE (Cai et al., 2017):** Selects the node which maximizes a linear combination of three metrics: PageRank centrality, model entropy and information density.
- **BALD (Gal et al., 2017; Houlsby et al., 2011):** Selects the node which has the the largest mutual information value between predictions and model posterior.
- **MetAL (Ours):** Selects the node maximizing the quantity in Equation (12)

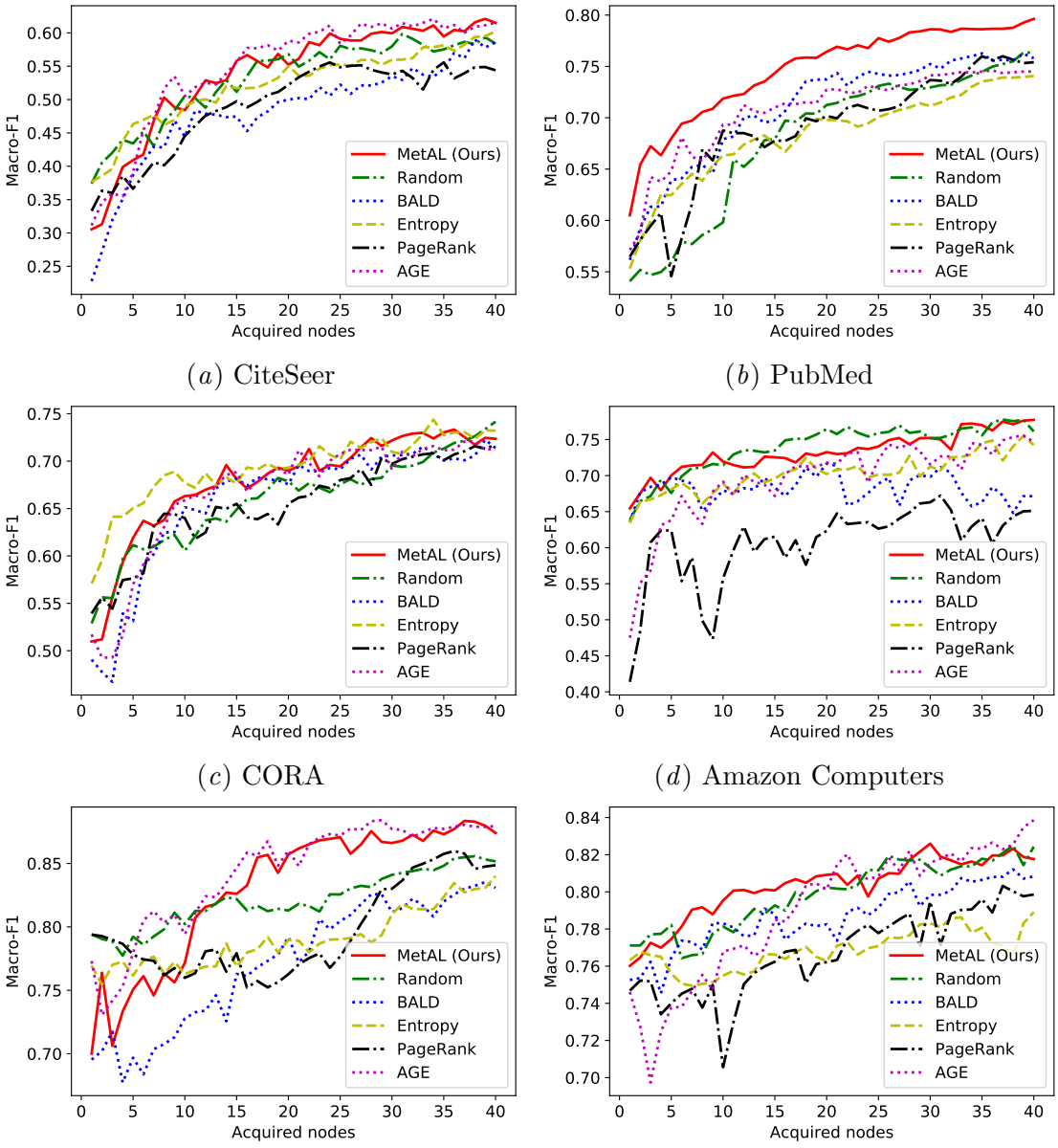
Here, entropy and BALD are uncertainty-based acquisition functions. For computing entropy, mutual information in BALD, and posterior class probabilities predicted by the current model $P(\hat{y}_q = k|G, Y_{\mathcal{L}})$ in MetAL, we use 20 iterations of MC-dropout to approximate a Bayesian model (Gal and Ghahramani, 2016). In contrast, centrality metrics such as PageRank and degree centrality can be considered as heuristics for selecting 'influential' instances in a graph dataset. The sequence of acquisitions is determined only based on the graph structure and does not depend on the features of instances nor the current set of labeled instances.

We acquire the label of an unlabeled node and retrain the GNN model by performing 50 steps of adam optimizer (Kingma and Ba, 2014). We perform 40 acquisition steps and repeat this process on 10 different randomly initialized training and test splits for each dataset. We report the average F1 score (Macro-averaged) over the test sets in each experiment. In most cases, average accuracy follows a similar trend. In MetAL, we execute 10 steps of gradient descent with momentum as the inner optimization loop and then we calculate the meta-gradient matrix.

4. Results and Discussion

4.1. Comparison of AL Strategies

In Figure 2 we observe that MetAL contributes to the best performance when the GCN model is used as the node classifier. We do not show the performance for degree centrality sampling for the clarity of visualizations since it exhibits the worst performance compared to all other acquisition functions. Figure 3 shows that MetAL works similarly with SGC as the classifier as well. However, we observe that the performance of SGC on some datasets is inferior compared to the GCN model. Lack of a hidden layer and nonlinear activation

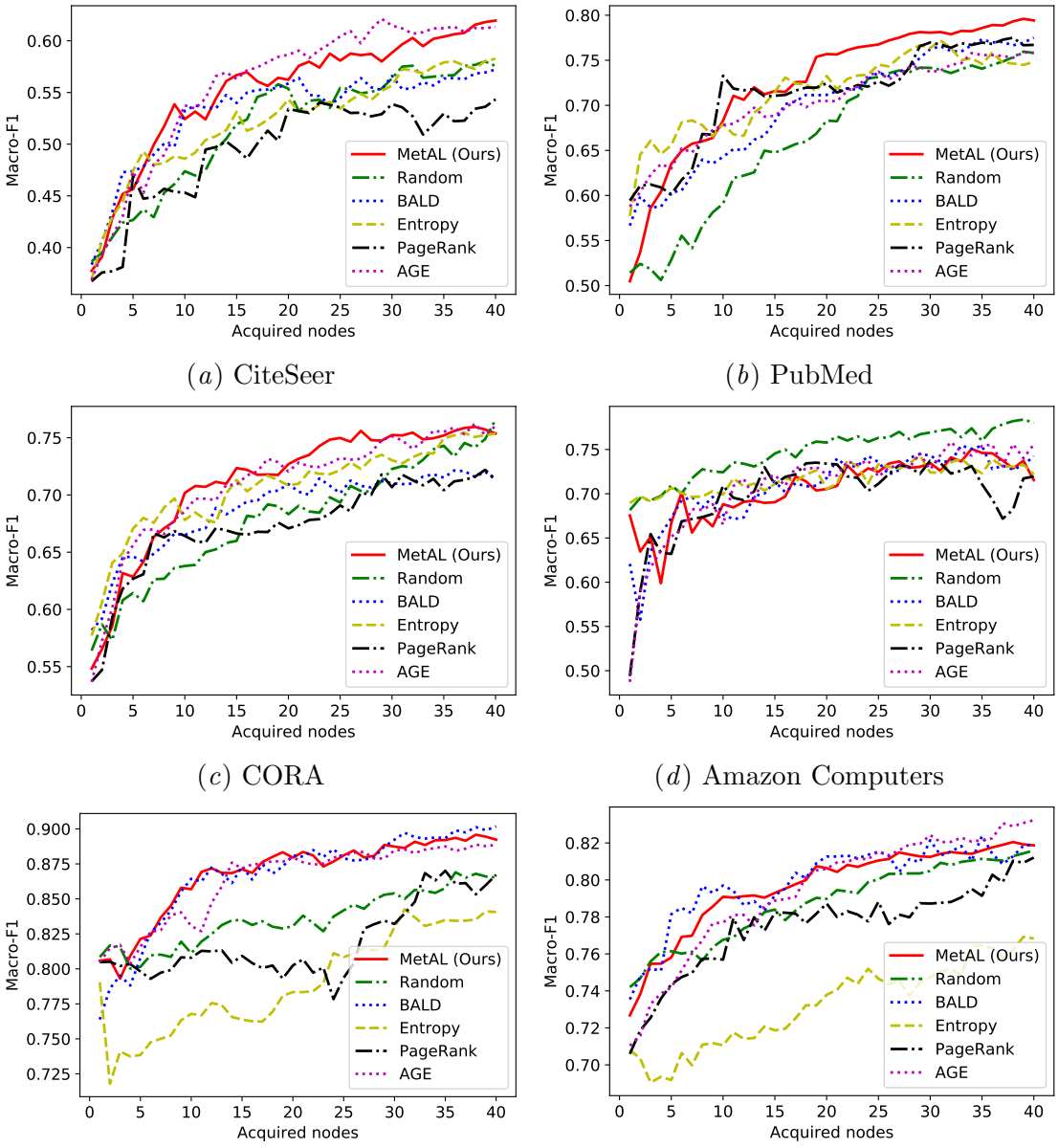


(a) CiteSeer (b) PubMed

(c) CORA (d) Amazon Computers

(e) Co-author Physics (f) Co-author CS

Figure 2: Performance of active learners with a 2-layer GCN model as the node classifier. Macro-F1 score (test) of active learning algorithms with number of acquisitions.

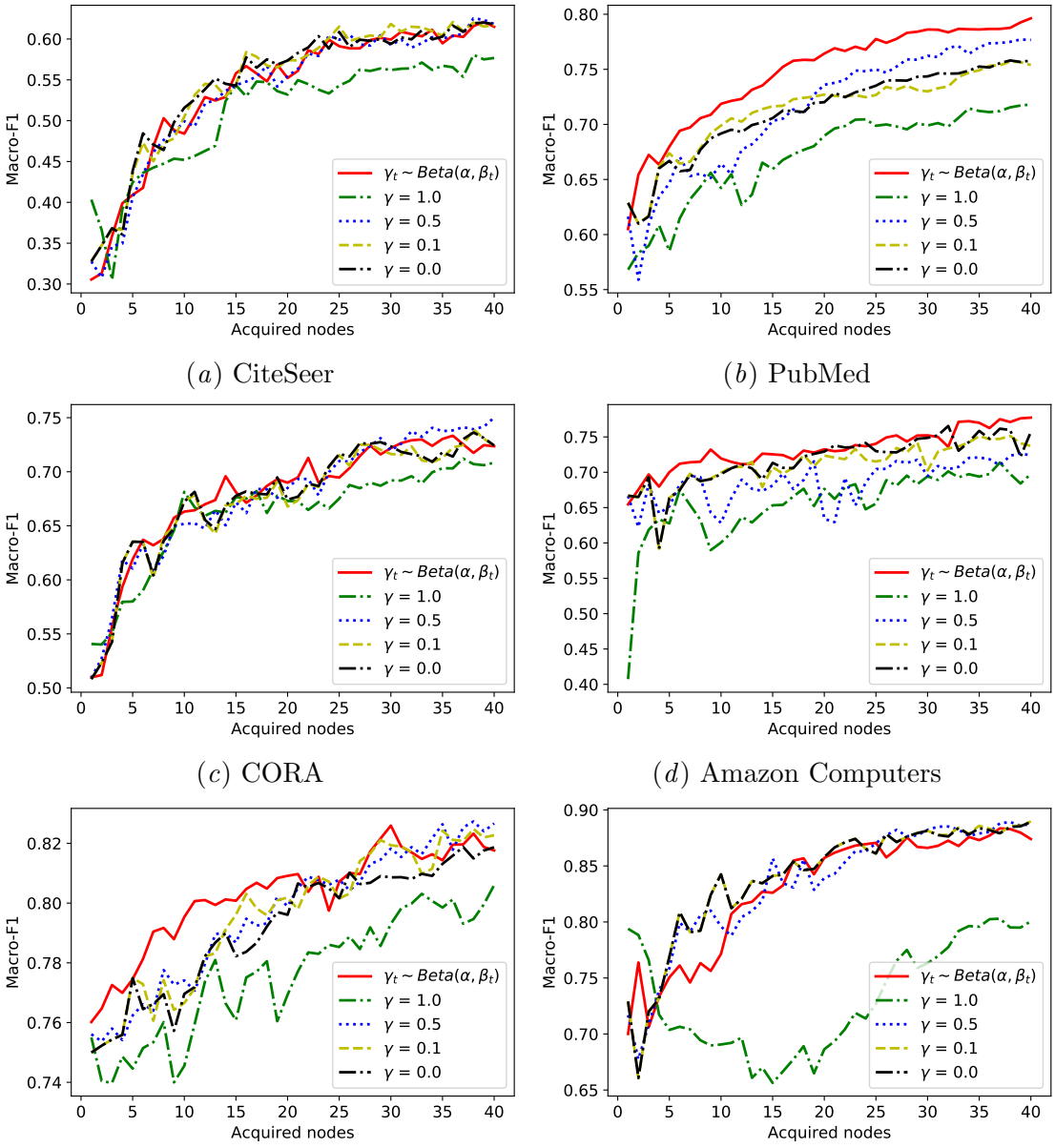


(a) CiteSeer (b) PubMed

(c) CORA (d) Amazon Computers

(e) Co-author Physics (f) Co-author CS

Figure 3: Performance of active learners with an SGC model as the node classifier. Macro-F1 score (test) of active learning algorithms with number of acquisitions. .



(*a*) CiteSeer (*b*) PubMed

(*c*) CORA (*d*) Amazon Computers

(*e*) Co-author CS (*f*) Co-author Physics

Figure 4: The importance of exploration coefficient γ. Our algorithm MetAL is run on these datasets with different values of γ: 0.1, 0.1, 0.5, and 1.0. The performance of MetAL with a fixed γ value is compared against the performance when it is sampled from a Beta distribution which is time-dependent.

functions can be the reason contributing to reduced performance. Even though PageRank centrality is proposed as a heuristic for acquiring nodes of a graph in previous work (Cai et al., 2017), we observe that its performance is inferior on larger graphs such as Amazon and co-authorship graphs. The performance of uncertainty-based active learners (entropy and BALD) is not consistent over different datasets. It is interesting that MetAL consistently outperforms AGE, the graph-specific AL benchmark without relying on time-consuming clustering algorithms. As one of its constituent criteria, AGE computes an information density measure using the learned features of the GNN model. This step depends on clustering the unlabeled instances and then calculating the euclidean distance to the cluster centers. This process is time consuming as evident in Table 2.

Figure 4 shows the results of the ablation studies we perform to understand the impact of the exploration coefficient γ_t. Here, we run the acquisition step in Equation (12) with different γ_t values: 0, 0.1, 0.5, and 1.0. Time-dependent γ_t sampled from a Beta distribution works the best on most datasets. Notably, selecting nodes solely based on the meta-gradient values ($\gamma_t = 0$) results in competitive results in most cases. However, pure exploration ($\gamma_t = 1$) results in inferior performance. This demonstrates that our proposed meta-gradient criterion is successful in finding 'informative' instances for labeling. However, this experiment shows that performance can be further improved by adaptively updating γ_t based on the feedback of the active learner.

4.2. Running Time

Table 2 lists the execution time each algorithm spends to acquire a set of 40 unlabeled instances on average. Even though our proposed approach MetAL consumes additional time compared to uncertainty-based algorithms, it is several times faster than the graph-specific baseline AGE. For example, MetAL is 20 times faster than AGE for the co-author Physics dataset. Further, the ultimate goal of applying AL is to reduce total human time spent on labeling instances. MetAL achieves this key objective at the cost of slightly increased acquisition time.

5. Related Work

5.1. Graph Neural Networks (GNNs)

GNNs (Li et al., 2015; Kipf and Welling, 2017; Wu et al., 2019) achieve state-of-the-art performance on the node classification problem providing a significant improvement over previously used embedding algorithms (Perozzi et al., 2014; Yang et al., 2016). What sets GNNs apart from previous models is their ability to jointly model both structural information and node attributes. In principle, all GNN models consist of a message passing scheme that propagates feature information of a node to its neighbors. Most GNN architectures use a learnable parameter matrix for projecting features to a different feature space. Usually, two or more of such layers are used along with a nonlinearity (e.g. ReLU). With normalized adjacency matrix $\hat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ a two-layer GCN model (Kipf and Welling, 2017) can be expressed as

$$Y_{\text{GCN}} = \text{softmax}\left(\hat{A}\ \text{ReLU}\left(\hat{A}X\theta^{(0)}\right)\theta^{(1)}\right), \tag{13}$$

Table 2: Running time (in seconds): average time taken to acquire 40 unlabeled instances. We run all experiments on a single Nvidia GTX 1080-Ti GPU.

Classifier	Dataset	Random	Entropy	PR	AGE	BALD	MetAL
GCN	CiteSeer	4.2	4.8	4.8	21.5	4.8	9.7
	PubMed	6.9	7.6	25.4	1125.9	7.9	34.6
	CORA	4.2	4.5	4.6	26.8	4.5	9.8
	Co-author CS	20.4	22.3	40.8	2154 .2	23.7	61.3
	Co-author Phy.	46.1	50.5	116.4	2436.9	50.8	125.4
	Amazon Comp.	17.5	19.1	31.8	1688.9	19.2	45.2
SGC	CiteSeer	1.7	1.9	2.1	18.3	1.9	5.4
	PubMed	2.0	2.2	20.0	1229.2	2.2	30.6
	CORA	1.3	1.8	1.8	23.7	1.9	5.5
	Co-author CS	16.8	19.8	33.2	2098.2	19.8	48.6
	Co-author Phy.	35.6	40.7	90.4	2232.3	40.8	97.0
	Amazon Comp.	12.2	12.5	17.2	1134.6	12.5	22.0

where $\tilde{A}$ and $\tilde{D}$ are the adjacency matrix and the degree matrix of graph G. $\theta^{(0)}$ and $\theta^{(1)}$ are the weight matrices of two neural layers.

Wu et al. (2019) arrived at a much simpler model named SGC by removing hidden layers and nonlinear activations in GCN model. This model can be written as

$$Y_{\text{SGC}} = \text{softmax}\left(\hat{A}^k X\theta\right). \tag{14}$$

5.2. Active Learning

AL research has contributed a multitude of approaches for training supervised learning models with less labeled data. We recommend Settles (2009) for a detailed review of AL.The objective of most existing AL approaches is to select the most informative instance for labeling. Uncertainty sampling is the most commonly used AL approach. Gal and Ghahramani (2016) propose using dropout at evaluation time as a way to calculate the model uncertainty of convolutional neural networks (CNN). Gal et al. (2017) provide a comparison of various acquisition functions for quantifying the model uncertainty of CNN models. The use of meta-learning for AL has been considered in a few recent works (Woodward and Finn, 2017; Bachman et al., 2017). However, these algorithms are designed for the few-shot learning setting and tied to RNN-based meta-learning models such as matching networks (Vinyals et al., 2016). Additionally, their reliance on reinforcement learning makes the training difficult. In contrast, our approach is built on model agnostic meta-learning (MAML) (Finn et al., 2017) which is efficient and can be used with a variety of supervised loss functions.

6. Conclusion

In this paper we introduced MetAL, a principled approach to perform active learning on graph data. We expressed the semi-supervised active learning problem as a bilevel optimization problem and demonstrated that meta-gradients can be used to make the bilevel optimization problem tractable. Empirical performance on benchmark attributed graphs drawn from

multiple domains shows that our proposed method is superior to existing heuristics-based AL algorithms. We further show the importance of performing exploration in addition to exploitation in AL problems. Adaptively learning the exploration coefficient using the feedback from the active learner is an interesting future direction.

In this work, we acquire a single unlabeled instance in each AL step and retrain the classifier. However, acquiring a batch of instances can make the learning process more efficient by reducing the number of retraining steps. We consider adapting MetAL for batch-mode acquisition as another avenue for future improvement. Additionally, understanding which characteristics of an attributed graph make AL easier or difficult is an open research problem. Such an understanding will lead to more efficient AL algorithms in the future.

Acknowledgments

This work was supported by JSPS Grant-in-Aid for Scientific Research(B) (Grant Number 17H01785) and JST CREST (Grant Number JPMJCR1687).

References

Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. In *International Conference on Learning Representations (ICLR)*, 2020.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Philip Bachman, Alessandro Sordoni, and Adam Trischler. Learning Algorithms for Active Learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 301–310. JMLR. org, 2017.

Mustafa Bilgic, Lilyana Mihalkova, and Lise Getoor. Active Learning for Networked Data. In *Proceedings of the 27th International Conference on Machine Learning*, pages 79–86, 2010.

Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. Active Learning for Graph Embedding. *arXiv preprint arXiv:1705.05085*, 2017.

Benoît Colson, Patrice Marcotte, and Gilles Savard. An Overview of Bilevel Optimization. *Annals of operations research*, 153(1):235–256, 2007.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1126–1135. JMLR. org, 2017.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1568–1577, 2018.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48, pages 1050–1059, 2016.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian Active Learning with Image Data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1183–1192. JMLR. org, 2017.

Li Gao, Hong Yang, Chuan Zhou, Jia Wu, Shirui Pan, and Yue Hu. Active Discriminative Network Representation Learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2142–2148. AAAI Press, 2018.

Quanquan Gu and Jiawei Han. Towards Active Learning on Graphs: An Error Bound Minimization Approach. In *2012 IEEE 12th International Conference on Data Mining*, pages 882–887. IEEE, 2012.

Yuhong Guo and Dale Schuurmans. Discriminative Batch Mode Active Learning. In *Advances in Neural Information Processing Systems*, pages 593–600, 2008.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian Active Learning for Classification and Preference Learning. *arXiv preprint arXiv:1112.5745*, 2011.

Ming Ji and Jiawei Han. A Variance Minimization Criterion to Active Learning on Graphs. In *Artificial Intelligence and Statistics*, pages 556–564, 2012.

Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, 2017.

Pang Wei Koh and Percy Liang. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1885–1894. JMLR. org, 2017.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated Graph Sequence Neural Networks. *arXiv preprint arXiv:1511.05493*, 2015.

Oisin Mac Aodha, Neill DF Campbell, Jan Kautz, and Gabriel J Brostow. Hierarchical Subquery Evaluation for Active Learning on a Graph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 564–571, 2014.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based Recommendations on Styles and Substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52, 2015.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online Learning of Social Representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–710. ACM, 2014.

Nicholas Roy and Andrew McCallum. Toward Optimal Active Learning through Monte Carlo Estimation of Error Reduction. *Proceedings of the 18th International Conference on Machine Learning*, pages 441–448, 2001.

Neil Rubens, Mehdi Elahi, Masashi Sugiyama, and Dain Kaplan. Active Learning in Recommender Systems. In *Recommender Systems Handbook*, pages 809–846. Springer, 2015.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective Classification in Network Data. *AI magazine*, 29(3):93–93, 2008.

Burr Settles. Active Learning Literature Survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

Mark Woodward and Chelsea Finn. Active One-shot Learning. In *Deep Reinforcement Learning Workshop, NeurIPS 2016*, 2017. URL https://arxiv.org/abs/1702.06559.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying Graph Convolutional Networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6861–6871. PMLR, 2019.

Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, pages 40–48, 2016.

Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377, 2019.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD 18, pages 974–983. Association for Computing Machinery, 2018.

Ye Zhang, Matthew Lease, and Byron C Wallace. Active discriminative text representation learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Daniel Zügner and Stephan Günnemann. Adversarial Attacks on Graph Neural Networks via Meta Learning. In *International Conference on Learning Representations*, 2019.