Active Learning on Graphs via Meta Learning

Kaushalya Madhawa [1] **Tsuyoshi Murata** [1]

Abstract

Active learning (AL) for semi-supervised node classification aims to reduce the number of labeled instances by selecting only the most informative nodes 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. We propose MetAL, an AL algorithm that selects unlabeled items that directly improve the future performance of a graph neural network (GNN) model. We formulate the AL problem as a bilevel optimization problem. Based on recent work in meta-learning, we compute the meta-gradients to approximate the impact of unlabeled instances on the model uncertainty. We empirically demonstrate that MetAL outperforms existing AL algorithms.

1. Introduction

The performance of a classification model depends on the size and the quality 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 instances based on an informative metric, gets their labels, and updates the labeled dataset. Then the classification model is retrained using the acquired labeled instances. This process is repeated until a good performance (e.g. accuracy) is reached.

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). However, such heuristics do not directly optimize the expected future performance of the model. Even if a heuristic works well on one dataset may not necessarily translate to improved performance on a different dataset. Therefore, it is desirable to directly incorporate model performance into the acquisition function instead of designing problem-specific heuristics.

Node classification is a semi-supervised learning problem. The learning algorithm can utilize all data instances including unlabeled ones. Only the labels of unlabeled instances are not known. Graph neural network (GNN) models (Li et al., 2015; Kipf & Welling, 2017; Wu et al., 2019a) have achieved state-of-the-art results in node classification (Wu et al., 2019b). However, the performance of a GNN model depends on a significant number of labeled nodes, as training and validation sets. In this paper, we study the problem of applying AL for classifying nodes of 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) by selecting only the most informative nodes for labeling.

Instead of designing heuristics, we build our work motivated by the framework of *expected error reduction* (EER) (Roy & McCallum, 2001; Guo & Schuurmans, 2008; Mac Aodha et al., 2014), in which the objective is to query instances which would minimize the generalization error. We formulate this objective as a bilevel optimization problem. Based on recent advances in meta-learning (Finn et al., 2017), we utilize meta-gradients to make this optimization efficient. Zügner & 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 5, with empirical evidence, we show that MetAL significantly outperforms existing AL algorithms.

Our contributions are: (1) MetAL, a novel active learning algorithm based on the expected error reduction principle; and (2) demonstrating that MetAL can consistently outperform existing baselines on a variety of real world graphs.

[1]Department of Computer Science, Tokyo Institue of Technology, Japan. Correspondence to: Kaushalya Madhawa <kaushalya@net.c.titech.ac.jp>.

GRL+ Workshop at 37^{th} International Conference on Machine Learning, Vienna, Austria.

2. Related Work

2.1. 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 & 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) provides 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 & 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.

2.2. Active Learning for Graph Data

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 & Han, 2012; Bilgic et al., 2010; Ji & 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. As we show in this paper, the performance of such models is inconsistent; efficient on some datasets, worse than random sampling on other datasets.

3. Our Framework

3.1. 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 $V_{\mathcal{L}}$ are given and labels of rest of the nodes $V_{\mathcal{U}}$ are unknown. A labeled node is assigned a label in $\{1, 2, \dots C\}$, where C is the number of classes. The objective of a learner is to learn a function $f_\theta(x_i)$ which can predict the class label of a given test node $i \in V_{\mathcal{U}}$. 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 $V_{\mathcal{L}}$ is much smaller compared to a large pool of unlabeled items $V_{\mathcal{U}}$. We can acquire the label of any unlabeled item by querying an oracle 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 number of K items. An optimal active learner selects the set of K items which would maximize the performance 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 a batch of B items are queried and the model is retrained with their labels.

3.2. Optimization Problem

We define our objective as finding q unlabeled instances which maximizes the likelihood of labeled instances while minimizing the uncertainty of labels 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 (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

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

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

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 contributes to the smallest expected value of $\mathcal{E}$. $\hat{\theta}^{+(x_q,y_q=k)}$ denotes the parameters of a model trained with instance q having label k.

3.3. 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. Additionally, this approach requires training a separate model for each possible label of each unlabeled item ($N_{\mathcal{U}} \times C$). Training such a large number of models is prohibitively time-consuming.

To remedy this issue, we estimate the impact of a query q with 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.

We rewrite Equation (1) as

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

We quantify the impact of retraining the model with (x_q, y_q) added to the labeled set as the change in loss $\Delta\mathcal{E}_{q,k} = \mathcal{E}(f_{\hat{\theta}_{x_q, y_q=k}}) - \mathcal{E}(f_{\hat{\theta}})$ and the expected change of loss for querying the item q by

$$\Delta\mathcal{E}_q = \sum_{k=1}^{C} P(\hat{y}_q = k | G, Y_{\mathcal{L}})\ \Delta\mathcal{E}_{q,k}. \quad (6)$$

$P(y_q = k|G, Y_{\mathcal{L}})$ is the posterior class probabilities of the current model $f_{\hat{\theta}}$. When $\delta_{q,k}$ is arbitrarily small, this change can be computed as the gradient of loss with respect to 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 (6) using gradient as

$$q^* = \arg\min_{q} \sum_{k=1}^{C} P(\hat{y}_q = k | G, Y_{\mathcal{L}})\ \nabla_{\delta_{q,k}} \mathcal{E}(f_{\hat{\theta}_{x_q, y_q=k}}). \quad (7)$$

The term $\Delta\mathcal{E}_q$ quantifies the impact of labeling a query q. This simplifies the AL problem to finding the item corresponding to the minimum expected meta-gradient $\Delta\mathcal{E}_q$ (Equation (7)) such that a negative 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 gradient ($-\Delta\mathcal{E}_q$).

Equation (5) and Equation (7) form a bilevel optimization problem. Calculating the meta-gradients as in Equation (7) involves a 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). 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 (7) 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

$$\hat{\theta}^{+(x_{\mathcal{Q}}, \hat{Y}_{\mathcal{Q}} \odot (1+\delta))} = \arg\min_{\theta} l(f_\theta(G), Y_{\mathcal{L}} \cup \hat{Y}_{\mathcal{Q}} \odot (1+\delta)). \quad (8)$$

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 with its label to the labeled set and retrain the model.

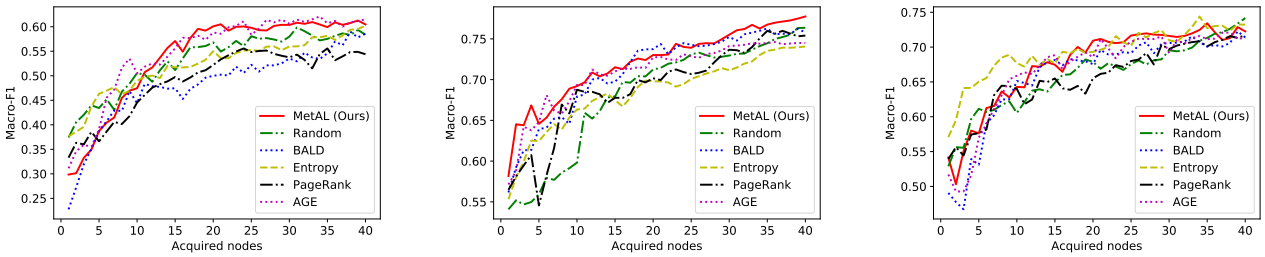
4. Experiments

4.1. Data

We evaluate our proposed approach on three citation network datasets: Citeseer, PubMed, and CORA (Sen et al., 2008). Details of the datasets can be found in the Supplementary Material. As the initial labeled set $V_{\mathcal{L}}$, we randomly select two nodes belonging to each label. 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.

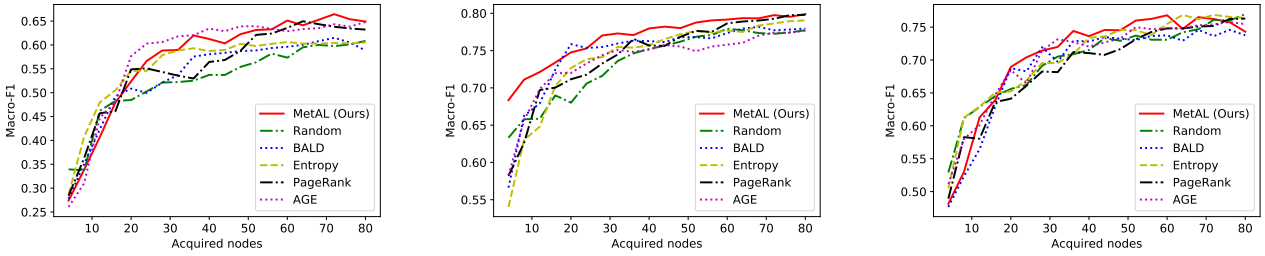
4.2. Model

We evaluate the effectiveness of MetAL, the proposed algorithm using a two-layer GCN model (Kipf & Welling, 2017). We use the default hyper-parameters used in GNN literature (e.g. learning rate = 0.01) and 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 following algorithms in our comparison:



(a) CiteSeer (b) PubMed (c) CORA

Figure 1. Macro-F1 score (test) of active learning algorithms with number of acquisitions. A two-layer GCN is used as the GNN model.



(a) CiteSeer (b) PubMed (c) CORA

Figure 2. Batch acquisition with GCN. Macro-F1 score (test) of active learning algorithms with number of acquisitions. A batch of 4 instances are acquired at each acquisition step.

- **Random:** Selects B unlabeled nodes randomly.
- **PageRank:** Select unlabeled nodes with B-largest PageRank centrality values.
- **Entropy:** Calculate the entropy of predictions of the current model over unlabeled nodes and select B nodes corresponding to B-largest entropy values.
- **AGE (Cai et al., 2017):** Selects B nodes which maximizes a linear combination of three metrics: PageRank, model entropy and information density.
- **BALD (Gal et al., 2017; Houlsby et al., 2011):** Selects B nodes which has the B-largest mutual information values between predictions and model posterior.
- **MetAL:** Selects B items corresponding to the minimum meta-gradient according to Equation (7)

For computing entropy, mutual information in BALD, and 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 & Ghahramani, 2016).

We execute 10 steps of gradient descent with momentum as the inner optimization loop and then calculate the meta-gradient matrix. We acquire the label of unlabeled items and retrain the GNN model by performing 50 steps of adam optimizer (Kingma & 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 respective test sets. In most cases, average accuracy follows a similar trend.

5. Results

In Figure 1 we observe that MetAL contributes to the best performance with both GCN and SGC models used as node classifier. We observe that the performance of SGC is inferior compared to the GCN model. Lack of a hidden layer and non-linear activation functions can be the reason contributing to reduced performance. The performance drop is noticeable, especially on the PubMed dataset. Additionally, we perform batch-mode acquisition: acquiring labels of a batch of 4 nodes at each step. Figure 2 shows that MetAL is the best AL algorithm for all three datasets.

6. Conclusion

We introduce MetAL, a principled approach to perform active learning on graph data. We express the semi-supervised active node classification problem as a bilevel optimization problem. Empirical performance on benchmark graphs shows that our proposed method is superior to existing heuristics-based AL algorithms. Understanding which characteristics of an attributed graph makes AL easier or difficult is an open research problem.

References

Ash, J. T., Zhang, C., Krishnamurthy, A., Langford, J., and Agarwal, A. Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. In *International Conference on Learning Representations (ICLR)*, 2020.

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

Bilgic, M., Mihalkova, L., and Getoor, L. Active Learning for Networked Data. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 79–86, 2010.

Bojchevski, A. and Günnemann, S. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. In *International Conference on Learning Representations*, pp. 1–13, 2018.

Cai, H., Zheng, V. W., and Chang, K. C.-C. Active Learning for Graph Embedding. *arXiv preprint arXiv:1705.05085*, 2017.

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

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

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

Gao, L., Yang, H., Zhou, C., Wu, J., Pan, S., and Hu, Y. Active Discriminative Network Representation Learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 2142–2148. AAAI Press, 2018.

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

Guo, Y. and Schuurmans, D. Discriminative Batch Mode Active Learning. In *Advances in Neural Information Processing Systems*, pp. 593–600, 2008.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. Bayesian Active Learning for Classification and Preference Learning. *arXiv preprint arXiv:1112.5745*, 2011.

Ji, M. and Han, J. A Variance Minimization Criterion to Active Learning on Graphs. In *Artificial Intelligence and Statistics*, pp. 556–564, 2012.

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

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

Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. Gated Graph Sequence Neural Networks. *arXiv preprint arXiv:1511.05493*, 2015.

Mac Aodha, O., Campbell, N. D., Kautz, J., and Brostow, G. J. Hierarchical Subquery Evaluation for Active Learning on a Graph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 564–571, 2014.

Namata, G., London, B., Getoor, L., Huang, B., and EDU, U. Query-driven Active Surveying for Collective Classification. In *10th International Workshop on Mining and Learning with Graphs*, volume 8, 2012.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

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

Rubens, N., Elahi, M., Sugiyama, M., and Kaplan, D. Active Learning in Recommender Systems. In *Recommender Systems Handbook*, pp. 809–846. Springer, 2015.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. Collective Classification in Network Data. *AI magazine*, 29(3):93–93, 2008.

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

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching Networks for One Shot Learning. In *Advances*

in Neural Information Processing Systems, pp. 3630–3638, 2016.

Woodward, M. and Finn, C. Active One-shot Learning. *arXiv preprint arXiv:1702.06559*, 2017.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., and Weinberger, K. Simplifying Graph Convolutional Networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 6861–6871. PMLR, 2019a.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. A Comprehensive Survey on Graph Neural Networks. *arXiv preprint arXiv:1901.00596*, 2019b.

Yao, L., Mao, C., and Luo, Y. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 7370–7377, 2019.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. 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, pp. 974983, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3219890. URL https://doi.org/10.1145/3219819.3219890.

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

Supplementary Material

A. Algorithm

We present the proposed algorithm here.

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 with the current model
  Sample a set of $N_{\mathcal{Q}}$ instances $\mathcal{Q}_j$ from $\mathcal{U}$
  Train a model with perturbed labels of $\mathcal{Q}_j$ instances with Equation (8)
  Calculate meta-gradient $\nabla_{\delta_{\mathcal{Q}_1}}$
  Select the best instance q^* using Equation (7)
  Query instances q^* and retrieve its 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

B. Model Details

We implement all algorithms in Pytorch (Paszke et al., 2019) and perform all experiments on a single Nvidia GTX 1080 GPU.

In addition to the GCN model, we perform the same experiments with SGC (Wu et al., 2019a), a simplified GNN architecture which does not include a hidden layer and non-linear activation functions. In Figure 3, we observe that the performance of SGC is slightly inferior compared to the GCN model. Lack of a hidden layer and non-linear activation functions can be the reason contributing to reduced performance.The performance drop is noticeable, specially on the PubMed dataset.

C. Data

We consider the largest connected component as an undirected graph in our experiments.

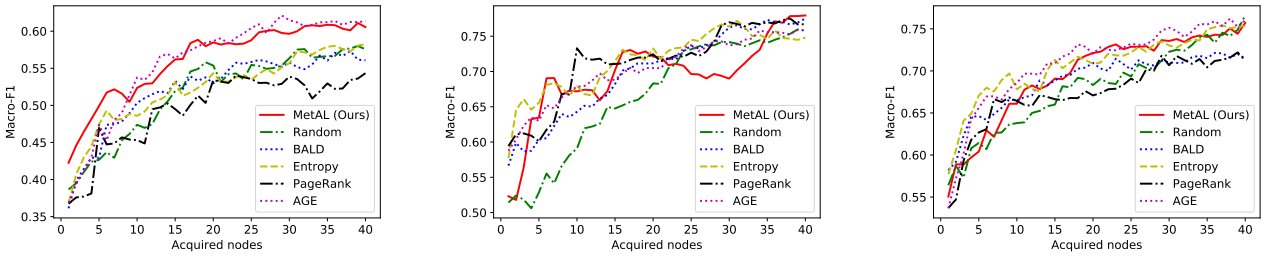
Citation Graphs. Each of these dataset is made of documents as nodes and edges between them. If one document cites another, they are linked by an edge. Each node contains bag-of-word features of its text as its attributes.

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

Dataset	N_V	N_C	$\lvert V_{\mathcal{L}}\rvert$ (%)
CiteSeer	2110	6	12 (0.56)
PubMed	19717	3	6 (0.03)
CORA	2485	7	14 (0.56)

D. 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 magnitudes faster than the graph-specific baseline AGE. Further, the ultimate goal of applying AL is to reduce total human time spent on labeling instances. It is safe to say that MetAL achieves this key objective at the cost of slightly increased acquisition time.



(a) CiteSeer (b) PubMed (c) CORA

Figure 3. Macro-F1 score (test) of active learning algorithms with number of acquisitions. A two-layer GCN is used as the GNN model.

Table 2. Running time (seconds): average time taken to acquire 40 unlabeled instances. Run on a single Nvidia GTX 1080 GPU.

Classifier	Dataset	Random	Entropy	PageRank	AGE	BALD	MetAL
GCN	CiteSeer	12.8	15.3	13.4	50.3	15.4	39.5
	PubMed	24.2	28.2	65.9	2312.9	28.3	193.2
	CORA	12.3	14.6	13.5	61.2	14.6	41.4
SGC	CiteSeer	4.7	5.0	5.6	41.0	5.1	25.0
	PubMed	5.0	8.1	48.8	2219.2	8.0	164.3
	CORA	3.8	4.8	5.8	55.0	4.9	27.8