Active Learning with Multi-Granular Graph Auto-Encoder

Yi He*, Xu Yuan*, Nian-Feng Tzeng*, Xindong Wu†‡
* Center for Advanced Computer Studies, University of Louisiana at Lafayette
† Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
‡ Mininglamp Academy of Sciences, Mininglamp Technology
*{yi.he1, xu.yuan, tzeng}@louisiana.edu, †xwu@hfut.edu.cn
Corresponding author: xu.yuan@louisiana.edu

***Abstract*—Predictive modeling of networked data finds many real-world applications, such as fraud detection in social networks, drug discovery in biomedical networks, paper topic classification in citation networks, and so forth. Although the advanced machine learning approaches can help build reasonably accurate predictive models, their applicability is immensely hindered by the data labeling tasks, which are onerous, time-consuming, and error-prone. In this paper, we propose a novel active learning paradigm for networked data, named *topology-and-content-aware* (TACA) active learning, aiming to minimize the number of labels while achieving a desirable level of model accuracy. Overall, TACA advances existing works from two aspects: (1) TACA makes no assumption on the network property, whereas most existing works only perform effectively on a locally consistent network in which linked nodes are expected to share the same labels and (2) TACA generates queries without relying on model performance, thereby enjoying robust predictive results even when noises exist in the queried labels. Both theoretical and empirical evidences are presented, substantiating the effectiveness of and optimism our approach.**

***Index Terms*—Active Learning, Networked Data, Node Embedding Techniques, Graph Neural Networks**

I. Introduction

Supervised learning systems such as deep learning usually envision a large labeled training dataset, so as to achieve leading performance in real-world applications. Data labeling in practice, however, is often a long, laborious, and expensive process, making the deployment of such systems economically impractical. To overcome this issue, *active learning* (AL) has been proposed with the aim of relieving labeling overhead. The key design of AL is to craft a querying strategy that wisely decides which instances to ask an oracle (*e.g.*, human expert) for their labels. Both theoretical [1], [2] and empirical evidences [3]–[5] have substantiated that, being assisted with a well-crafted strategy, the learning system can attain a desired level of accuracy at a significantly reduced labeling budget.

Despite effective, most existing AL methods are tailored for data instances that are identically and independently distributed (*i.i.d.*). However, such an *i.i.d.* nature may not hold in many applications, where the data instances are linked through topological structures. Examples abound, *e.g.*, social networks where cyber users are connected via follower-followee relations [6], [7], bibliographic graphs with research papers citing or cited by, other papers to form nature links [8], [9], e-commercial networks where online products being frequently bought together are correlated [10], [11]. We refer to such data characterizable by graphs as the *networked data*.

The challenge of performing AL with networked data lies in the lack of a principled querying strategy for *harmonizing* such information that is conveyed by instance features and graph topologies. Existing works fail to address this challenge well and are limited in two aspects. First, to leverage the graph information, prior works make the *local consistency* assumption on the network property, where the linked instances are highly expected to share the same labels [12]–[15]. As a result, they do not generalize well to a wider range of problems in which such an assumption does not hold.

Second, most AL methods generate label-querying strategies in a model-focused means. That is, to select the next instance, they must depend on the performance of the learner trained so far, presuming the labels of the previously queried instances are perfectly correct. Unfortunately, as a general case, human labelers can make mistakes. Being trained on noisy labels, the learners suffer from deteriorated performance, leading to non-informative queries in the remaining iterations, where the new labels cannot improve model performance or may even increase the prediction errors [16]–[18].

Motivated by these observations, this work explores two questions: (1) Can the network information be more effectively exploited without relying on the local consistency assumption? (2) Can we craft a querying strategy that is *model-free*, so as to enjoy an inherent robustness to noisy labels?

Our answer to these questions provides a novel active learning paradigm, named *topology-and-content-aware* (TACA) active learning. The key idea is twofold: (1) learning informative node embeddings without making any assumption on the network property, and (2) generating queries with no dependence on model performance. To realize the idea, the design of TACA comprises two main phases. In phase **I**, we devise a novel *multi-granular graph auto-encoder* (MGAE) to embed the instance features and network topologies into a latent space in a harmonic manner [19], [20]. Specifically, MGAE takes into account both i) local-consistency-based knowledge, in which immediate neighbors on graphs tend to have identical labels, and ii) global-consistency-based knowledge, in which instances having similar feature semantics are likely to share the same labels. Notably, MGAE operates in a purely unsupervised fashion, thus able to tolerate noisy labels.

In phase **II**, based on the node embeddings learned in

phase I, we generate queries by employing the uncertainty reduction principle [18], so as to abort any reliance on model performance. In particular, we theoretically show that the instances with the highest level of predictive uncertainty, on which the learner is most likely to make errors, can be identified via gauging their geometric relations with other node embeddings in the latent space. Through labeling these most uncertain instances, the learner become immune to the same type of errors, thereby improving prediction accuracy.

Due to the page limitation, we defer technical details (*e.g.*, proofs) and complete experiments to supplementary material by the following link: https://bit.ly/3d1nS0m.

II. Active Learning with Networked Data

Problem Statement. Let a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represent the networked data, where each edge $E_{ij} \in \mathcal{E}$ links two nodes V_i and V_j, and each node $V_i \in \mathcal{V}$ represents an instance being associated with a d-dimensional feature vector $\mathbf{x}_i \in \mathbb{R}^d$.

Let $h^* : \mathcal{V} \mapsto \mathcal{Y}$ be the true hypothesis underlying data, where $\mathcal{Y} \in \{Y_1, \ldots, Y_C\}$ denotes a label space comprising C classes. We aim to learn a predictive model f that can approximate h^* with a minimized labeling effort. Let $\mathcal{L}$ and $\mathcal{U}$ denote the labeled and unlabeled nodes (instances), respectively, and $\mathcal{V} \equiv \{\mathcal{L}, \mathcal{U}\}$. Our learning problem is then formulated as a min-max game as follows.

$$\begin{aligned} &\min d_{\mathcal{H}}(h^*, f), \quad s.t. \ \ |\mathcal{L}| \leq B, \\ &f = \arg\max_{f \in \mathcal{H}} \mathbb{E}_{V_i \in \mathcal{L}}[\Pr(y_i \mid V_i, f)], \end{aligned} \tag{1}$$

where B denotes the labeling budget and $d_{\mathcal{H}}(\cdot, \cdot)$ represents a distance metric defined over the hypothesis space $\mathcal{H}$ that gauges the discrepancy between our learned model f and the true hypothesis h^*. Our goal is to search an f that approximates h^* within a small labeling budget $B \ll |\mathcal{V}|$.

A. Challenges and Prior Efforts

One plausible idea to tackle the label scarcity issue is to enrich the node information. Intuitively, the more information each node contains, the smaller the number of labels required to train an accurate model would be. For networked data, the information sources from two channels – the node features and the network topology. We desire a node embedding method that can *harmonizes* both two channels of information, so as to better the model performance.

Crafting such an embedding method is non-trivial, where three challenges arise in our learning problem. *First*, because the labels are scarce (due to limited labeling budget) and noisy (due to human mistakes), this embedding method must operate in an unsupervised fashion, which remains an open and challenging problem in graphs. Existing methods [19], [21], [22] that entail label supervision for learning node embeddings are functionally infeasible in this context.

Second, real-world networked data are usually sizeable by comprising tens of thousands of nodes and hundreds of thousands of edges. The embedding method must scale-up to such large graphs. Conventional graph learning methods [23]–[26] requiring as an input the entire graph are not scalable and thus inapplicable. Also, these methods in general capture node cluster structures, which is linear, leading to restricted expressiveness hence inferior prediction accuracy.

Third, although recent graph-neural-network (GNN) advances [15], [27]–[29] have manifested remarkably performance by means of learning scalable and expressive node embeddings, they commonly make the local consistency assumption which, unfortunately, may not hold in real practices. Consider, for example, a fraudulent user in a social network who only follows (links to) multiple normal users on purpose. The local consistency assumption enforces a message passing among the linked nodes, such that they are placed closely in the embedding space. Thus, the malicious message (*e.g.*, advertising, rumoring) of the fraudulent users are "washed-out" by the benign features of their normal neighbors. As a result, such fraudulent users are over-confidently misclassified as normal, if its neighboring normal users are queried and labeled before. Such examples abound in a wide range of real applications [7], [30] to which, evidently, those GNN-based node embedding methods are not generalizable.

B. Our Insights

Tailoring an embedding method to overcome these challenges can provide active learners with informative node representations, leading to desirable learning performance. However, to our best knowledge, no existing research has addressed all the three challenges simultaneously.

To fill the gap, we propose a novel graph embedding method, termed multi-granular graph auto-encoder (MGAE), which possesses four nice properties as follows. i) It learns node embeddings in a purely unsupervised manner. ii) It uses neural architecture to capture non-linear data structures underlying the embedding space. iii) It does not take the entire graph as input; Rather, it can be updated in a stochastic way and hence is scalable. iv) It lifts the local-consistency assumption by leveraging the global-consistency network information, thereby being more applicable. These properties empower our MGAE method to well address the aforementioned challenges.

III. Our Approach

We now present the technical details of TACA active learning approach. Specifically, TACA consists of two core building blocks: Multi-granular graph auto-encoder that learns informative node embeddings from both local and global network neighbors (in Section III-A) and Model-free active query generator that decides the nodes to be labeled without relying on the model performance (in Section III-B).

A. Multi-Granular Graph Auto-encoder (MGAE)

The nature of designing MGAE is built upon a key observation that nodes conveying similar feature semantics can be topologically faraway. Revisit the example – a group of fraudulent users spreading malicious messages may not link to each other but only to the normal users on purpose. In such situations, the nodes associated with similar and informative feature contents scatter across the network and can be faraway from each other. It is thus desirable to learn node embeddings by taking such *global-consistency-based* knowledge into consideration. Our MGAE method is to serve this purpose.

Overall, MGAE consists of an encoder and a decoder. In the encoding phase, it discovers the community structure that underlies the node feature space, and then leverages such a structure to extract an intermediate graph in a coarse granularity. To clarify notion, the nodes in the coarse graph are termed *abstract nodes*. Each abstract node synopsizes the information from a node community, within which the nodes share the similar feature contents. In the decoding phase, MGAE inherits an inner-product-based read-out function from [27]; It reads out the coarse graph from the abstract nodes at first, and then it decomposes the abstract nodes into a finer granularity, based on which the original network is reconstructed.

By this design, each node embedding is forced to aggregate information from other nodes those in the same community (which could be topologically faraway) hence is less likely to be overridden by their immediate neighbors. The details of the encoder and decoder are presented as follows.

Encoder: coarse graph extraction. The operations of each encoder layer comprise two main steps. *First*, we explore the underlying node communities from the node feature matrix via spectral clustering, which equals to solving the nonnegative matrix factorization problem [31]. *Second*, we apply soft-assignments of nodes to generate the adjacency matrix and node embeddings, yielding coarse graph and abstract nodes, respectively. Specifically, consider an encoder with L hidden layers, where each layer is a convolution layer interleaved with a graph coarsening operation, as shown in Figure 1. The output of the l-th convolution layer is recursively given by:

$$\mathbf{Z}^{(l)}_{\text{temp}} = \text{ReLU}\left(\tilde{\mathbf{A}}^{(l-1)}\mathbf{Z}^{(l-1)}\mathbf{W}^{(l)}\right) \in \mathbb{R}^{n_{l-1}\times z_l}, \quad (2)$$

where $l \in \{1,\dots,L\}$ and $\mathbf{Z}^{(0)} = \mathbf{X} \in \mathbb{R}^{n\times d}$ is the original node feature matrix. Denoted by $\tilde{\mathbf{A}}^{(0)} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}} \in \mathbb{R}^{n\times n}$ the normalized graph Laplacian, with $\mathbf{A}$ being the adjacency matrix of the original network topology. The weight matrix $\mathbf{W}^l \in \mathbb{R}^{z_{l-1}\times z_l}$ parameterizes the layer and $\mathbf{W}^1 \in \mathbb{R}^{d\times z_1}$. As such, $\mathbf{Z}^{(l)}_{\text{temp}}$ lowers the dimension of each node vector but does not change the number of nodes in the graph. The graph coarsening operation then reduces the number of nodes by constructing abstract nodes, defined as:

$$\mathbf{Z}^{(l-1)} \approx \mathbf{P}^{(l)}\mathbf{Q}^{(l)\top} \in \mathbb{R}^{n_{l-1}\times z_{l-1}} \quad (3)$$

$$\mathbf{Z}^{(l)} = \mathbf{P}^{(l)\top}\mathbf{Z}^{(l)}_{\text{temp}} \in \mathbb{R}^{n_l\times z_l}, \quad (4)$$

$$\tilde{\mathbf{A}}^{(l)} = \mathbf{P}^{(l)\top}\tilde{\mathbf{A}}^{(l-1)}\mathbf{P}^{(l)} \in \mathbb{R}^{n_l\times n_l}, \quad (5)$$

where $\mathbf{Q}^{(l)} \in \mathbb{R}_+^{z_{l-1}\times n_l}$ represents the basis matrix, with its column vectors being the cluster centroids and there are n_l clusters in total. $\mathbf{P}^{(l)}$ indicates the cluster membership matrix in which each entry $\mathbf{P}^{(l)}_{i,j}$ indicates the probability that node V_i belongs to the j-th cluster.

To understand what is happening in each encoder layer, we extrapolate the *physical meanings* of Eqs. (2), (3), (4), and (5) as follows. Consider the first layer that takes in the original node feature vectors $\mathbf{X} \in \mathbb{R}^{n\times d}$. First, Eq. (2) propagates the feature information of each node to its first-order neighbors, mapping the node vectors in a new latent space: $\mathbb{R}^d \mapsto \mathbb{R}^{z_1}$. And, in a parallel fashion, the node communities underlying the original $\mathbb{R}^d$-feature space is discovered via Eq. (3). Second,

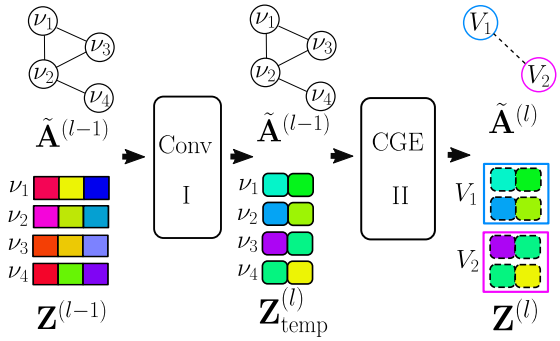


Fig. 1: Illustration of two main steps in the encoding phase. (I) Convolution layer changes the dimension of the node embeddings $z_{l-1} \mapsto z_l$; (II) Coarse graph extractor reduces the number of nodes in the network $n_{l-1} \mapsto n_l$.

the nodes falling into the same community are aggregated to form the abstract nodes with linear combinations through Eq. (4), and the connectivity among those abstract nodes indicates the topology of the coarse graph, calculated in Eq. (5). An illustration of this process is shown in Figure 1.

By stacking more such encoder layers, local-consistency-based knowledge, which is afforded by the convolution layers, and global-consistency-based knowledge, which is captured through the construction of the coarse graphs, are embedded in an interchangeable manner.

Decoder: hierarchical readout. Our decoder proceeds a hierarchical readout process, starting from the most coarse-grained graph, decomposing its abstract nodes to a finer granularity, hence reading-out a finer-grained graph. This process iterates until the original network is recovered.

Specifically, the number of decoder layers equals to that in the encoder, *i.e.*, L. The hierarchical readout in each decoder layer is defined as follows.

$$\hat{\mathbf{A}}^{(l+1)} = \phi\left(\left(\mathbf{P}^{(l)}\right)^{\dagger}\mathbf{Z}^{(l)}\mathbf{Z}^{(l)\top}\left(\mathbf{P}^{(l)\top}\right)^{\dagger}\right), \quad (6)$$

where $\phi(\cdot)$ is the logistic sigmoid and $(\cdot)^{\dagger}$ denotes the generalized Moore-Penrose pseudo inverse [32]. Along this way, the last decoder layer outputs a recovery of the original network topology, *i.e.*, $\hat{\mathbf{A}}^{(L)} \in \mathbb{R}^{n\times n}$. The entire MGAE architecture is updated according to the discrepancy between $\hat{\mathbf{A}}^{(L)}$ and the input network topology $\tilde{\mathbf{A}}$, *e.g.*, $\ell(\tilde{\mathbf{A}}, \hat{\mathbf{A}}^{(L)}) = \|\tilde{\mathbf{A}} - \hat{\mathbf{A}}^{(L)}\|_{\text{F}}^2$, where $\ell(\cdot,\cdot)$ is a differentiable and desirable convex loss function such as the Frobenious-norm loss.

B. Model-Free Active Query Generator

Based on the learned node embedding, we now study how to generate active queries for predictive modeling. The key idea lies to select the nodes on which the model is most likely to make errors. By labeling these most *uncertain* nodes, the model can avoid making the same type of errors, thereby enjoying an improved prediction performance.

However, the human labors are likely to make mistakes in practice, yielding noisy labels. The model trained with these noises is distorted. Traditional querying strategies that are *model-focused*, *e.g.*, [33], [34], depending on historical model performance to generate new queries, are hence negatively affected by the distorted model; Indeed, they are likely to

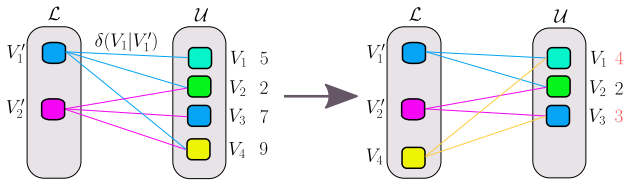


Fig. 2: A bipartite system view of our active querying strategy.

generate non-informative queries since it is not distinguishable whether an uncertain instance carries unseen and new knowledge or is merely labeled by mistake.

To combat against this issue, we in this work leverage the recent theoretical advance [18] to design a *model-free* active query generator, aborting the reliance on model performance by relating the node uncertainty with the node-wise geometric property in the embedding space. Specifically, we cast the node querying task into a bipartite system optimization problem, where labeled and unlabeled nodes are disjointly placed at the two sides of the system. At each iteration, one node from the unlabeled side is moved to the labeled side by following the *overall uncertainty reduction* principle.

Let $\delta(V_i \mid V_j)$ denote the uncertainty of node V_i when node V_j is labeled. To select the node that maximizes the uncertainty reduction for the entire system, we solve the program below.

$$\arg\max_{V_i^* \in \mathcal{U}} \sum_{\substack{V_i \in \mathcal{U} \\ V_j \in \mathcal{L}}} \delta(V_i \mid V_j) - \sum_{\substack{V_i \in \mathcal{U} \setminus V_i^* \\ V_j \in \mathcal{L} \cup V_i^*}} \delta'(V_i \mid V_j) \,, \tag{7}$$

where the first term sums up the uncertainty between a pair of two nodes, each of which from one side of the system. The second term evaluates the overall system uncertainty after a node V_j^* is moved from $\mathcal{U}$ to $\mathcal{L}$, *i.e.*, labeled.

Figure 2 visualizes how a node is queried by optimizing Eq. (7). There are two labeled nodes V_1' and V_2' and four unlabeled nodes V_1, V_2, V_3, and V_4. After querying V_4, the overall uncertainty reduction of the system comprises two parts. First, the uncertainty of V_4 has been directly removed (*i.e.*, $9 \rightarrow 0$). Second, by knowing V_4, the uncertainties of V_1 and V_3 are intermediately reduced (*i.e.*, $5 \rightarrow 4$ and $7 \rightarrow 3$, respectively). Thus, the overall uncertainty reduction of querying V_4 is $9 + 1 + 4 = 14$. Similarly, we calculate overall uncertainty reduction individually for V_1, V_2 and V_3, and select the node having the maximal value.

The problem then is to calculate the node uncertainty, with the crux lying in how to relate the model accuracy with the nodes' geometric relations. Fortunately, we can exploit the Maximum Distance Minimization (MDM) theory [18] to derive an ℓ_1-distance-based uncertainty measurement, which is computationally simple yet offers our querying strategy a nice property as follows.

Theorem III.1 (MDM)**.** *Let $\mathbf{z}_i$ and $\mathbf{z}_j$ denote the embeddings of an arbitrary unlabeled node V_i and a labeled node V_j, respectively. Denoted by*

$$\Delta(V_i) = \sum_{V_j \in \mathcal{L}} \delta(V_i \mid V_j) := \inf_{V_j \in \mathcal{L}} \|\mathbf{z}_i - \mathbf{z}_j\|_1 \tag{8}$$

the overall uncertainty of V_i given a labeled set $\mathcal{L}$. The estimation error $d_{\mathcal{H}}(h^*, f)$ is upper bounded by node V_i that maximizes Eq. (8), namely $V_i = \arg\max_{V_i \in \mathcal{U}} \Delta(V_i)$.

Theorem III.1 suggests that, through querying the label of a node that maximizes the infimum ℓ_1-distance to the previously labeled nodes, the discrepancy upper-bound between our learned model (*i.e.*, f) and the true hypothesis (*i.e.*, h^*) is lowered in an iterative greedy fashion. As a result, the node queried in such a manner is guaranteed to be informative, where the gain of model accuracy can be reasonably expected.

IV. Experiments

This section delivers empirical evidence to exhibit that our topology-and-content-aware (TACA) active learning approach is effective and can achieve superior performance over the active learning counterparts. To be specific, we elaborate the general evaluation setups in Section IV-A and present the experimental results in Section IV-B.

A. General Setup

Datasets. Our experiments are carried out over five real-world datasets that are widely used in the graph learning literatures. The details are depicted as follows.

Cora, *DBLP*, and *PubMed* are three bibliographic datasets that are widely used in the literature [8], [9], [12], representing the citation networks. Each node in the citation network indicates one research paper, which may cite or be cited by other papers, naturally forming the links between pairs of nodes. In addition, each node is also described by a feature vector, with entries representing the unique words used in the respective paper. The class labels corresponding to each dataset represent one domain of interest. For example, the class labels in the Cora dataset represent 7 machine-learning-related domains, including neural networks, probabilistic methods, genetic algorithms, and others.

Amazon [10] is a dataset profiling the co-purchase network from Amazon, in which nodes represent products, edges pinpoint the product pairs that are frequently bought together, node features include bag-of-words embedded in product reviews, and class labels indicate the product category. *Flickr* [11], [35] is an image dataset collected from Flickr, in which two nodes (images) are linked if they share common metadata, such as from the same location, submitted to the same gallery, and taken by friends. Each image is associated with a feature vector (*e.g.*, reviews) to encode the image information. The class labels are given by the image tags. Table I summarizes the detailed statistics of the five datasets.

Evaluation Protocol. We split each dataset into training/validation/test sets with an $8:1:1$ ratio. Before learning, the labels of all instances in the training set are masked. At each iteration, once one query is generated, the label of the corresponding instance is revealed. The predictive model is trained on the instances whose labels have been exposed so far, and then performs prediction on the test set. To avoid the cold-start problem, we start building the predictive model when at least 20 instances have been quired.

In MGAE, one hidden layer is implemented for both the encoder and the decoder. That is, one intermediate coarse graph is extracted from the original data network, and then, based on the abstract nodes, the topologies of the coarse

TABLE I: Statistics of the studied datasets

Dataset	# of Nodes	# of Edges	# of Features	# of Classes
Cora	2708	10,556	1433	7
DBLP	17,716	105,734	1639	4
PubMed	19,717	88,648	500	3
Amazon	7487	119,043	745	8
Flickr	89,250	899,756	500	7

graph and the original network are reconstructed in a layer-by-layer fashion. The width of the hidden layer determines the network aggressiveness of compressing information from the feature vectors. The number of the abstract nodes decides the information grainy degree of the discovered node communities – the larger the number, the more fine-grained information is captured by each node community. We tune these parameters via the validation set, with the hidden layer width and the abstract node number chosen from $\{128, 64, 32, 16, 8\}$ and $\{11, 9, 7, 5, 3\}$, respectively. For predictive modeling, we employ an MLP (multi-layer perceptron) with one hidden layer as the classifier. Adam optimizer is employed for training both the encoder and the classifier in a full-batch manner, with an ℓ_2-penalized weight decay of $5\text{e}{-4}$. The learning rate is grid-searched in $\{1\text{e}{-5}, 1\text{e}{-4}, 1\text{e}{-3}\}$.

Compared Methods. For comprehensive comparison, we take several representative active learning methods that are graph-theory-based, optimization-based, and deep-learning-based as counterparts, which are highlighted below.

ALTG [36] generates instance queries purely based on the geometric properties of graphs. Specifically, this method acquires labels of the nodes that disconnect most regions of a graph through minimal edge cuts.

PAL [5] formalizes active learning under a Markov decision framework. Deep reinforcement learning is employed to learn an optimal policy that maximizes model performance with a fixed labeling budget.

ALFNET [12] trains a content-only learner and a collective learner. It pre-clusters the data instances, and iteratively generates queries from the cluster that yields the highest level of disagreement between the two learners.

LLGC [13] casts the active learning problem into the problem of minimizing transductive Rademacher complexity. To leverage network information, this method regularizes the predictive model by imposing the local consistency assumption, *i.e.*, the linked instances sharing the similar labels.

B. Experimental Results

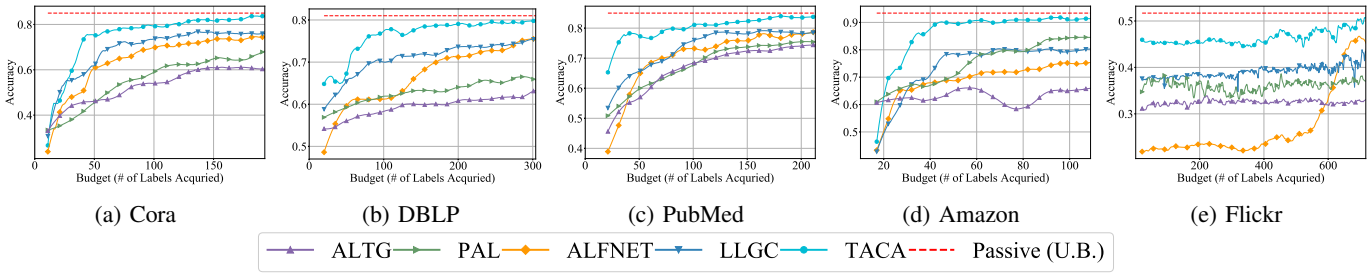
Figure 3 shows the classification accuracy of our TACA approach and the compared counterparts on all datasets under varying budget (*i.e.*, the number of labeled instances). Notably, if we use the labels of all training instances to directly train a predictive model, the performance of the obtained predictive model indeed offers an upper bound of the active learning methods, as indicated by the dashed red line in Figure 3. From these figures, the learning accuracy of every method is seen to keep increasing with a larger budget. The reason is that the predictive model can enjoy more information provided by more labeled instances to achieve better prediction performance. However, our TACA expedites the learning curve of classifier more effectively and is more accurate than all other methods with the same limited budget across all datasets. Such results confirm that TACA can select more informative instances comparing to other active learning methods. In other words, TACA achieves accurate prediction performance with the fewest queries among all compared methods.

In addition, from Figures 3 (a), (b), (c), (d), and (e), we observe that ALTG almost performs the worst for all datasets. This demonstrates that it is insufficient to rely only on the graph geometric properties to generate meaningful queries in attributed graphs, implying that information resided in the node feature space directly determines model performance and shall not be ignored. PAL performs the second worst on 4 out of 5 datasets. This suggests that information conveyed by the node features is also insufficient for node classification, necessitating to consider graph topologies. One exception occurs on the Amazon dataset, where PAL beats ALTG, ALFNET, and LLGC. This is probably due to the reason that local consistency may not hold in the Amazon product classification tasks. For example, the frequently bought products (*e.g.*, the add-on items) may link with multiple other products from very different categories. Enforcing the linked products to share similar labels/features may hinder model representativeness (consider the canonical beer-diaper example), which thus degrades the prediction performance. This finding is also supported by the underperformed ALFNET and LLGC on the Amazon dataset, since both methods explicitly make the local consistency assumption.

Moreover, comparing ALFNET to LLGC, we observe that the learning speed of ALFNET is extremely slow, although it achieves almost similar results as those of LLGC after convergence, *i.e.*, ALFNET ties LLGC on Cora, DBLP, and PubMed but it outperforms LLGC on Flickr. These results confirm better the robustness of the model-free active learning methods (*i.e.*, LLGC and TACA) than that of the model-focused method (*i.e.*, ALFNET). Specifically, if noises exist in the previously queried labels, the trained model is inaccurate. As a result, the model-focused method fails to gauge the informativeness of the unlabeled instances, thus suffering from randomness until enough queried instances are correctly labeled. Flickr is such a noisy dataset, where the passive learner can achieve only 51% accuracy. ALFNET is seen to underperforms all the compared methods with the limited budget of less than 600 labeled instances. In contrast, our proposed TACA does not rely on model performance when generating queries hence performs robustly on Flickr even with a few labels.

V. Conclusion

This paper proposed a *topology-and-content-aware* (TACA) active learning paradigm towards accurate predictive modeling for networked data without onerous labeling effort. The key idea of our paradigm is to learn informative node embeddings, such that the active learner can enjoy enriched node information and hence achieves good prediction performance with a few node labels. To that end, a novel embedding method, named multi-granular graph auto-encoder (MGAE),



(a) Cora (b) DBLP (c) PubMed (d) Amazon (e) Flickr

Fig. 3: Classification accuracy of ALTG, PAL, ALFET, LLGC, and our TACA on all five datasets with various budgets.

is proposed to *harmonize* information conveyed by both node features and network topologies. MGAE advances the prior art by taking into account the knowledge with respect to both local and global network consistencies, without imposing any assumption on the network property. A *model-free* querying strategy was then crafted to query the labels for the most uncertain nodes by gauging their geometric relations in the embedding space. Notably, our querying strategy does not rely on the historical model performance and thus is robust to noisy labels. A theoretical analysis substantiated the effectiveness of our approach, and extensive empirical evaluation on five real-world datasets further evidenced the performance superiority of our approach over four active learning counterparts.

Acknowledgement

We thank ICDM 2020 reviewers for their constructive feedback. This work was supported in part by the National Key Research and Development Program of China under grant 2016YFB1000901 and the National Natural Science Foundation of China (NSFC) under grant 91746209, and in part by the US National Science Foundation (NSF) under grants 1652107, 1763620, and 1948374. Any opinion and findings expressed in the paper are those of the authors and do not necessarily reflect the view of funding agency.

References

[1] Z. Allen-Zhu, Y. Li, A. Singh, and Y. Wang, "Near-optimal design of experiments via regret minimization," in *ICML*, 2017, pp. 126–135.
[2] S. Hanneke *et al.*, "Theory of disagreement-based active learning," *Foundations and Trends® in Machine Learning*, vol. 7, no. 2-3, pp. 131–309, 2014.
[3] B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009.
[4] Y. Fu, X. Zhu, and B. Li, "A survey on instance selection for active learning," *Knowledge and information systems*, vol. 35, no. 2, pp. 249–283, 2013.
[5] M. Fang, Y. Li, and T. Cohn, "Learning how to active learn: A deep reinforcement learning approach," in *EMNLP*, 2017.
[6] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time url spam filtering service," in *S & P*. IEEE, 2011, pp. 447–462.
[7] B. Wang, N. Z. Gong, and H. Fu, "Gang: Detecting fraudulent users in online social networks via guilt-by-association on directed graphs," in *ICDM*. IEEE, 2017, pp. 465–474.
[8] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.
[9] A. Bojchevski and S. Günnemann, "Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking," in *ICLR*, 2018.
[10] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *SIGIR*, 2015, pp. 43–52.
[11] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. Prasanna, "Graph-saint: Graph sampling based inductive learning method," in *ICLR*, 2020.
[12] M. Bilgic, L. Mihalkova, and L. Getoor, "Active learning for networked data," in *ICML*, 2010, pp. 79–86.
[13] Q. Gu and J. Han, "Towards active learning on graphs: An error bound minimization approach," in *ICDM*. IEEE, 2012, pp. 882–887.
[14] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *ICLR*, 2014.
[15] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," in *ICLR*, 2018.
[16] Z. Lu, X. Wu, and J. Bongard, "Adaptive informative sampling for active learning," in *SDM*. SIAM, 2010, pp. 894–905.
[17] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 17–37, 2016.
[18] H. Zhang, S. Ravi, and I. Davidson, "A graph-based approach for active learning in regression," in *SDM*. SIAM, 2020, pp. 280–288.
[19] L. Gao, H. Yang, C. Zhou, J. Wu, S. Pan, and Y. Hu, "Active discriminative network representation learning," in *IJCAI*, 2018, pp. 2142–2148.
[20] C. Zhuang and Q. Ma, "Dual graph convolutional networks for graph-based semi-supervised classification," in *WWW*, 2018, pp. 499–508.
[21] J. Liang, P. Jacobs, J. Sun, and S. Parthasarathy, "Semi-supervised embedding in attributed networks with outliers," in *SDM*. SIAM, 2018, pp. 153–161.
[22] Z. Yang, W. W. Cohen, and R. Salakhutdinov, "Revisiting semi-supervised learning with graph embeddings," in *ICML*, 2016, pp. 40–48.
[23] S. Si, D. Shin, I. S. Dhillon, and B. N. Parlett, "Multi-scale spectral decomposition of massive graphs," in *NeurIPS*, 2014, pp. 2798–2806.
[24] K. Rohe, T. Qin, and B. Yu, "Co-clustering directed graphs to discover asymmetries and directional communities," *Proceedings of the National Academy of Sciences*, vol. 113, no. 45, pp. 12 679–12 684, 2016.
[25] T. Guo, S. Pan, X. Zhu, and C. Zhang, "Cfond: consensus factorization for co-clustering networked data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 4, pp. 706–719, 2018.
[26] Y. He, S. Chen, T. Nguyen, B. A. Wade, and X. Wu, "Deep matrix tri-factorization: Mining vertex-wise interactions in multi-space attributed graphs," in *SDM*. SIAM, 2020, pp. 334–342.
[27] T. N. Kipf and M. Welling, "Variational graph auto-encoders," in *NeurIPS*, 2017.
[28] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NeurIPS*, 2017, pp. 1024–1034.
[29] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
[30] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *KDD*, 2018, pp. 2847–2856.
[31] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *SDM*. SIAM, 2005, pp. 606–610.
[32] A. Ben-Israel and T. N. Greville, *Generalized inverses: theory and applications*. Springer Science & Business Media, 2003, vol. 15.
[33] K. Yu, J. Bi, and V. Tresp, "Active learning via transductive experimental design," in *ICML*, 2006, pp. 1081–1088.
[34] R. Chattopadhyay, Z. Wang, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, "Batch mode active sampling based on marginal probability distribution matching," in *KDD*, 2012, p. 741–749.
[35] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "Nus-wide: a real-world web image database from national university of singapore," in *ACM ICIVR*, 2009, pp. 1–9.
[36] N. Cesa-Bianchi, C. Gentile, F. Vitale, and G. Zappella, "Active learning on trees and graphs," *COLT*, 2013.