Active Learning on Heterogeneous Information Networks: A Multi-armed Bandit Approach

Doris Xin, Ahmed El-Kishky, De Liao, Brandon Norick, Jiawei Han
Department of Computer Science, The University of Illinois at Urbana-Champaign
{dorx0, elkishk2, deliao2, bnorick, hanj}@illinois.edu

***Abstract*—Active learning exploits inherent structures in the unlabeled data to minimize the number of labels required to train an accurate model. It enables effective machine learning in applications with high labeling cost, such as document classification and drug response prediction. We investigate active learning on heterogeneous information networks, with the objective of obtaining accurate node classifications while minimizing the number of labeled nodes. Our proposed algorithm harnesses a multi-armed bandit (MAB) algorithm to determine network structures that identify the most important nodes to the classification task, accounting for *node types* and without assuming label *assortativity*. Evaluations on real-world network classification tasks demonstrate that our algorithm outperforms existing methods independent of the underlying classification model.**

I. Introduction

As a collection of entities interconnected by *links*, a network aptly describes many systems with data dependencies, e.g. the Internet, academic collaborations, gene regulations. *Information networks* use the network construct to organize information, with links denoting channels for information exchange. We represent information networks as graphs with nodes representing entities and edges representing links between entities. One common task performed over information networks is classifying nodes by a function learned from existing node labels, e.g., classifying proteins, which form networks via interactions, by their biological functions.

Active learning addresses the label scarcity problem in classification tasks with high labeling cost but abundant unlabeled data. Instead of passively learning from a given set of labeled examples, the active learner examines unlabeled data and selectively queries for the labels of the most informative examples. Networks, with their rich structures, present an excellent opportunity for such studies. Due to the ubiquity of information networks in many scientific disciplines, active learning on information networks has the potential to drive the direction of future scientific research under the emergent concept of *data-driven science*. In the protein function example, the important proteins selected by the active learner can serve as guidance on the subject of new experimental studies.

Active learning on information networks requires many special considerations not applicable in a data independent setting (i.i.d,). Strategies that are effective for i.i.d., such as uncertainty sampling, fall short for networks where nodes are interdependent. Another major challenge is identifying the connection pattern between nodes in the same class, as many network classification tasks do not follow the *assortativity* assumption that nodes in the same class are more connected.

Previous studies on network active learning typically focus on networks containing a single type of nodes [2], [13]. In these *homogeneous* networks, strong assumptions about network structures are embedded in the way the links are constructed, limiting the range of analyses that can be performed. *Heterogeneous information networks* (HIN), which contain multiple types of nodes, enrich prior methods with relation semantics between the different types [18]. The HIN setting allows us to explore the relations that are pertinent to a given classification task, which is advantageous when the connectivity pattern between nodes belonging to different classes is unknown. Our algorithm does not assume assortativity nor dependency on the classification model, unlike existing work such as [21].

We propose MABAL (Multi-Armed Bandit for Active Learning), an adaptive active learning algorithm on HINs inspired by the *multi-armed bandit* problem. To ensure label coverage, we consider the *batch mode* setting in which multiples nodes are queried for labels at each iteration. The optimal sequence of query batches lies in an intractably large search space [5]. MABAL employs a combinatorial MAB (CMAB) algorithm to construct query batches based on the recommendations of simple heuristic strategies created from centrality rankings [4]. Each simple strategy represents a hypothesis about the network structure and the semantic relations that make a node an informative query, and MAB is used to handle the tradeoff between the exploration of all strategies and the exploitation of the current best strategies, in linear time w.r.t. the unlabeled data size, the batch size, and the number of simple strategies.

We conduct empirical evaluations of MABAL on multiple real world information networks, on multiple classification tasks over the same network, and with different classification models. While each of the heuristic and literature baselines succeeds only in limited capacities, MABAL consistently achieves high classification accuracy using fewer labels than baselines in all tasks and settings tested.

II. Preliminaries

We represent an information network as a graph $G = (V, E)$, where V is the set of nodes corresponding to the entities in the network and $E = \{e_{ij} = (v_i, v_j) \mid v_i, v_j \in V\}$ is the set of links between these entities. In a *heterogeneous*

information network, each node $v \in V$ is mapped onto a specific *type* $t \in T$ via $\tau : V \to T$. Let $V_t \subseteq V$ denote the set of nodes with type t. These subsets partition V, i.e., $V_t \cap V_{t'} = \emptyset \; \forall \; t \neq t' \in V$ and $\bigcup_{t \in T} V_t = V$.

Collective classification refers to the task of inferring the class label of all nodes based on the labels provided for a subset [15]. Let $\mathcal{Y}$ denote the set of class labels. Given the observations $\mathcal{L} = \{(v, y_v) \mid v \in V, y_v \in \mathcal{Y}\}$, our objective is to train a classifier f^* to approximate the hidden objective $f : V \to \mathcal{Y}$ that generated the node labels.

Graph Centrality Measures. Graph centrality measures quantify the importance of nodes in a network. For a node v, its *rank* in V under C is defined as $rank_{C|V}(v) = \sum_{v' \in V} I(v \leq_C v')$, i.e., the number of nodes with centrality greater than that of v. In other words, the most important nodes based on C have the lowest $rank_{C|V}$ values. In this work, we consider the following commonly used centrality measures: degree, closeness, betweenness, PageRank, eigenvector and Katz. We refer interested readers to [3] for an in-depth discussion on centrality measures.

Batch Mode Active Learning. Active learning can be conducted in many different modes based on the constraints of the specific application [16]. In this work we focus on *batch mode* learning in which the learner starts with a set of unlabeled instance, $\mathcal{U}$ ($\mathcal{U} \subseteq V$ in HINs), and issues *queries*, $Q \subseteq \mathcal{U}$, to the *oracle* $\mathcal{O}$, such as a human annotator. In this study, we assume that $\mathcal{O}$ provides for each query q a single label $y_q \in \mathcal{Y} \cup \emptyset$ ($\emptyset$ when the label is not available) and is *stable*, i.e., it always provides the same label for q whenever queried, which implies zero utility in re-querying the same instance. Batch mode active learning is parameterized by the *budget* B, the upper bound on the total number of queries, and the batch size b, the number of queries to issue at each iteration over $\lceil B/b \rceil$ iterations [16]. With the labels received from $\mathcal{O}$, the learner updates $\mathcal{U} = \mathcal{U} \setminus Q$, $\mathcal{L} = \mathcal{L} \cup Q$ and then its strategy for selecting future queries based on $\mathcal{L}$. Batch mode allows the learner to better adapt to the classification task from incremental changes in the observed label distribution. We assume uniform labeling cost over all instances in P. Works such as [12], [17], [20] address queries with varying costs.

Multi-armed Bandit. The multi-armed bandit (MAB) problem models the *exploration* v. *exploitation* tradeoff in sequential allocation tasks [9]. At each iteration i, a player makes a play p_i by pulling one of K arms on a *bandit* to receive a reward $r(p_i)$. The objective is to maximize the cumulative reward $R = \sum_{i=1}^{T} r(p_t)$ earned over T rounds. This is often modeled as minimizing a player's *regret*, the difference between R and R^{OPT} achieved by the optimal strategy. Each arm has a reward distribution, modeled as a random variable X_i. $\mu_i = \mathbb{E}[X_i]$ is unknown to the player, who must make decisions based on empirical rewards from past actions. Let $\bar{\mu}_k^t$ be the player's expectation of reward from arm k in round t. The player can either choose to *exploit* existing knowledge about the payoffs by playing $\text{argmax}_{k=\{1,\ldots,K\}} \bar{\mu}_k^t$ or to *explore* arms with high uncertainties. In *combinatorial multi-armed bandit* (CMAB), the player at each round plays multiple arms, or a *super arm* $S \in 2^{[K]}$, and receives feedback for S. We adopt this extension of the classic MAB problem for modeling query batches containing multiple nodes. In this work we make use of the algorithm proposed in [4], which assumes *semi-bandit* feedback, i.e., a reward is observed for individual arms in S.

III. The Algorithm

We introduce MABAL, our proposed active learning algorithm on HINs. The objective of MABAL is to find

$$\mathcal{Q}^* = \underset{\mathcal{Q} \in \mathcal{P}_B(\mathcal{U})}{\text{argmax}} \frac{1}{|\mathcal{U} \setminus \mathcal{Q}|} \sum_{v \in \mathcal{U} \setminus \mathcal{Q}} \mathbf{1}\{y_v = f_{\mathcal{Q}}(v)\} \qquad (1)$$

where y_v is the actual label for v, $f_{\mathcal{Q}}(v)$ is the label predicted by the classifier trained on $\mathcal{Q}$, and $\mathcal{P}_B$ is the powerset up to size B. MABAL adapts an existing CMAB algorithm, CUCB [4], to combine simple centrality-based strategies known as *primary candidate selections* (PCSs). CUCB estimates expected rewards based on the empirical rewards and the number of times an arm is explored. It boosts the reward expectations for under-explored arms to avoid dismissing a potentially optimal strategy without sufficient evidence.

While reminiscent of the query by committee (QBC) approach in active learning, where disagreement among an ensemble of hypotheses is used for query selection, MABAL significantly differs from QBC in that the PCSs do not propose any classification of the nodes, only an ordering on their importance in learning a given task. Instead of considering the label disagreement among experts, MABAL considers the ranking of the nodes as votes, which are then weighted by the rewards of the voters to create the query set. The only computation cost at each iteration is retraining a single model (as opposed to retraining a model for each hypothesis in QBC) since the node rankings can be precomputed in order to minimize delay between iterations.

A *primary candidate selector* (PCS), λ_t^C is constructed from the ordering induced on V_t by the centrality measure C. We denote the unlabeled set for type t as $\mathcal{U}_t = \mathcal{U} \cap V_t$. When queried in batch mode with batch size b, λ_t^C will return $Q^{\lambda_t^C} = \{v | v \in \mathcal{U}_t, rank_{C|\mathcal{U}_t}(v) \leq b\}$, i.e., the top b unlabeled nodes of type t with the highest centrality C. Note that λ_t^Cs with the same t share the same candidate pool $\mathcal{U}_t$ but prioritizes nodes in the pool differently for querying. $\mathcal{C}$, the set of centrality measures used in MABAL, can be any arbitrary combination of existing or novel centrality measures.

Let Λ be the set of PCSs in MABAL. To account for the possibility that none of the centrality measures serve as adequate active learning strategies, we add to Λ *Random* constructed from a random node ordering equivalent to a *passive* learner. Including *Random* in Λ also serves as a mechanism to regulate the active learner from overfitting to the centrality-based strategies. This prevents MABAL from doing worse than the passive learner when none of the centrality-based strategies prove to be effective.

A. Correspondence between AL and MAB

We define the *utility* of a query q given the labeled set $\mathcal{L}$ as: $u(q|\mathcal{L}) = R(f_{\mathcal{L}\cup q}) - R(f_{\mathcal{L}})$, where $f_{\mathbf{X}}$ is the classifier trained on the observations $\mathbf{X}$, and $R(f_{\mathbf{X}}) = \sum_{v\in V} \mathbf{1}\{y_v = f_{\mathbf{X}}(v)\}/|V|$ is the classification accuracy of $f_{\mathbf{X}}$ on V. The utility of $\mathcal{L}$ is simply $u(\mathcal{L}) = R(f_{\mathcal{L}})$. We can rewrite (1) as

$$\mathcal{L}^{OPT} = \underset{\mathcal{L}\in\mathcal{P}_B(V)}{\operatorname{argmax}}\, u(\mathcal{L}) \tag{2}$$

Suppose that $u(q|\mathcal{L})$ is obtainable after q is labeled. This establishes a correspondence between the objectives of AL and MAB as follows. Each arm in the MAB corresponds to a $\lambda \in \Lambda$. In the fully sequential setting, i.e., $b = 1$, some $\lambda \in \Lambda$ is picked to issue the query q_i at each iteration i, corresponding to a play p_i in MAB. The label budget B corresponds to T, the number of rounds played in MAB. After playing the sequence $(q_1, \ldots, q_T)$, we obtain $\mathcal{L} = \{q_1, \ldots, q_T\}$. $R(f_{\mathcal{L}})$ corresponds to the cumulative reward $\mathcal{R}$ in MAB earned by the sequence of plays $(q_1, \ldots, q_T)$. Thus, finding the sequence of plays that maximizes $\mathcal{R}$ is equivalent to (2). □

The simplest adaptation for $b > 1$ is to choose a single $\lambda \in \Lambda$ and use Q^λ as the query set Q. However, this strategy can severely reduce $\mathcal{L}$'s coverage of G and $\mathcal{Y}$. A classifier over G cannot be adequately learned without examples for some $y \in \mathcal{Y}$ or major components of G. We propose BATCH, an algorithm that computes Q by selectively taking *advice* from the PCSs based on their expected rewards $\{\bar{\mu}(\lambda)\}$. In brief, BATCH combines the advice Q^λ from λ weighted by its expected reward $\bar{\mu}(\lambda)$ to form Q in a way that also promotes diversity. We expound on BATCH in Section III-C.

In establishing the correspondence between AL and MAB, we assumed knowledge of the utility of query q, $u(q|\mathcal{L})$. However, $R(f)$ cannot be computed without the ground truth labels for all nodes, which are not available to the learner. Additionally, we need to define $u(Q|\mathcal{L})$ for the query set Q in batch mode. While it is tempting to think of $u(Q|\mathcal{L})$ as the sum of $u(q|\mathcal{L})$ for $q \in Q$, u is not additive due to data dependencies in a network. Consider a Q made up of two nodes v_1, v_2 connected by an edge. While v_1 and v_2 may have high utility individually, $u(Q)$ is not a sum of their utilities due to the coverage overlap between two adjacent nodes. We propose a novel network-based expected error reduction measure, ∇, as a proxy for u to address these issues. We define ∇ and $\hat{\mu}(\lambda)$, the *empirical* mean reward of λ, in Section III-B.

Algorithm 1 shows MABAL, with input information network G, label budget B, batch size b, and the set of PCSs Λ. $\hat{\mu}(\lambda)$ is the *empirical* reward for $\lambda \in \Lambda$, and T_λ is the number of nodes nominated by λ that were labeled. In line 8, we use CUCB to compute $\bar{\mu}(\lambda)$, the *expected* reward of λ used to compute the query set Q by `Batch`. We query the oracle $\mathcal{O}$ for the labels of nodes in Q and update $\mathcal{L}$, $\mathcal{U}$ and T_λs accordingly. To update the empirical rewards of the λs, we retrain the classifier on $\mathcal{L}$ and recompute $\hat{\mu}(\lambda)$ using (6).

Algorithm 1 MABAL

```
1: procedure MABAL(G = (V, E), B, b, Λ)
2:     i ← 0, L ← ∅, U ← V
3:     T_λ ← 0   ∀λ ∈ Λ                          ▷ Num. of queries from λ
4:     μ̂(λ) ← 1   ∀λ ∈ Λ                         ▷ Empirical reward for λ
5:     while i · b < B do
6:         b' = min(b, B − t · b)
7:         for all λ ∈ Λ do
8:             μ̄(λ) = μ̂(λ) + sqrt(3 ln i / (2 T_λ))
9:         Q = Batch(G, b', Λ, U, {μ̄(λ_C^t)})
10:        L ← L ∪ O(Q)
11:        U ← U \ Q
12:        T_λ ← T_λ + |Q ∩ Q^λ|   ∀λ ∈ Λ
13:        Train classifier on L
14:        update μ̂(λ)   ∀λ ∈ Λ using (6)
15:        i ← i + 1
16: return L
```

B. Entropy Reduction as MAB Reward

In the *expected error reduction* framework for active learning, queries are selected to minimize the generalization error over $\mathcal{U}$ [16]. Strategies in this framework are known to be computationally expensive, since finding the optimal query requires retraining on all possible queries. In batch mode, this becomes combinatorially more expensive.

For a given query q, the expected log-loss over $\mathcal{U}$ with θ trained on $\mathcal{L} \cup q$ is

$$\sum_{y\in\mathcal{Y}} P(y|q) \left(-\sum_{v\in\mathcal{U}} \sum_{y\in\mathcal{Y}} P_\theta(y|v) \log P_\theta(y|v) \right) \tag{3}$$

Since $H(\mathcal{Y}|v) = -\sum_{y\in\mathcal{Y}} P(y|v)\log(P(y|v))$ is the entropy in the label distribution of v, the above is equivalent to finding a query that minimizes the expected entropy over $\mathcal{U}$. The factor for marginalizing q over $\mathcal{Y}$ can be dropped since each query receives a single label from $\mathcal{O}$, i.e., (3) reduces to $\sum_{v\in\mathcal{U}} H(\mathcal{Y}|v)$. Based on the expected log-loss, we propose a graph-specific error reduction measure to distinguish queries by their ability to reduce generalization errors. At iteration i, let θ_i be the model trained on $\mathcal{L}$, which includes labels received from the oracle up to i. We define $\nabla_i(q)$, the local error reduction due to query q at time i as

$$\nabla_i(q) = \sum_{v\in N(q)} H_{\theta_{i-1}}(\mathcal{Y}|v) - H_{\theta_i}(\mathcal{Y}|v) \tag{4}$$

where $N(q)$ are the neighbors of q. For a query set Q, we have

$$\nabla_i(Q) = \sum_{v\in N(Q)} H_{\theta_{i-1}}(\mathcal{Y}|v) - H_{\theta_i}(\mathcal{Y}|v) \tag{5}$$

where $N(Q) = \bigcup_{q\in Q} N(q)$. $\nabla_i(Q) \neq \sum_{q\in Q} \nabla_i(q)$ when there are overlaps in the neighborhoods of nodes in Q. We use (4) to approximate $u(q|\mathcal{L})$, the reward for the query q, and (5) to approximate $u(Q|\mathcal{L})$, the reward for the query set Q. The reward of playing λ at time i is thus defined as $\nabla_i(\lambda) = \exp\left(\frac{\nabla_i(Q^\lambda)}{|\nabla_i(V)|} - 1\right)$. To avoid a large reduction in a single round from biasing the algorithm towards a particular λ for the remainder of the query budget, we normalize $\nabla_i(Q^\lambda)$ by $|\nabla_i(V)|$, the absolute value of total entropy reduction incurred in the ith iteration. We use the absolute value to avoid a false positive reward signal as an artifact of global entropy increase. We transform the ratio via $e^{(x-1)}$ so the reward for any λ is always positive and in the $[0, 1]$ range. The *empirical* reward mean for λ at time i can be computed as

$$\hat{\mu}_i(\lambda) = \frac{(i-1)\cdot\hat{\mu}_{i-1}(\lambda) + \nabla_i(\lambda)}{i} \tag{6}$$

C. Query Batch Selection

Algorithm 2 BATCH

procedure BATCH(G, b, Λ, $\mathcal{U}$, $\{\bar{\mu}_i(\lambda)\}$)
  $Q^* \leftarrow \bigcup_{\lambda \in \Lambda} Q^\lambda$
  $s_t \leftarrow 1 - \frac{b}{|V_{\tau(q)} \cap \mathcal{U}|+1} \quad \forall t \in T$
  $\mu'(\lambda) \leftarrow s_{\tau(\lambda)} \cdot H(\mathcal{Y}|\lambda) \cdot \bar{\mu}(\lambda) \quad \forall \lambda \in \Lambda$
  for all $q \in Q^*$ **do**
    $\bar{\mu}(q) \leftarrow \sum_{\lambda \in \Lambda(q)} \mu'(\lambda) \cdot v_\lambda(q)$
  $S = sort_{desc}(Q^*, \{\bar{\mu}(q)\})$
return $S[:b]$

Algorithm 2 shows BATCH, the subroutine in MABAL for computing the optimal query batch given the expected rewards of the PCSs. As an optimal query set should yield good coverage both in terms of the network and the classes in $\mathcal{Y}$, BATCH adjusts the expected rewards of the primary learners to account for these two factors via *type bias correction* s_t and *label diversity* $H(\mathcal{Y}|\lambda)$. Based on the adjusted PCS expectations, BATCH computes the expected reward of each query node as the sum of *votes* weighted by the rewards of the PCSs and selects the top b nodes with the highest votes as the query batch.

Expected Query Reward. The objective of `Batch` is to find a query set Q^{OPT} of size b that yields the highest reward expectation, i.e.,

$$Q^{OPT} = \underset{Q \in \mathcal{P}_b(\mathcal{U})}{\operatorname{argmax}} \mathbb{E}[u(Q|\mathcal{L})] \tag{7}$$

Since the exact value of the true maximum reward is not computable, we use (5) to search for Q^{OPT}. Computing Q^{OPT} then boils down to finding a query set resulting in the largest entropy reduction in its neighborhood (dependent on the underlying classification model). To be classification-model-agnostic, BATCH approximates expected entropy reduction via the expected rewards for the PCSs, which are functions of observed entropy reductions. For each $q \in Q^* = \bigcup_{\lambda \in \Lambda} Q^\lambda$, let $\Lambda(q)$ be the set of PCSs that selected q as a candidate. The expected reward of q under *weighted Borda count* is $\bar{\mu}(q) = \sum_{\lambda \in \Lambda(q)} \bar{\mu}(\lambda)(b - rank_{Q^\lambda}(q))$. Although $C(v) \in [0,1] \quad \forall v \in V, C \in \mathcal{C}$, the ranges of centrality measures can differ greatly. To counter any negative effects of this artifact, we use the *Borda count*, a positional voting system that allows for a fair comparison of the PCSs irrespective of their ranges [6]. We define the expected reward of Q as a function of $\bar{\mu}(\lambda)$ as

$$\bar{\mu}(Q) = \sum_{q \in Q} \sum_{\lambda \in \Lambda(q)} \bar{\mu}(\lambda) \cdot \bar{\mu}(q) \tag{8}$$

which assumes additivity of query rewards, namely, if a node is in the neighborhood of multiple queries, the effects of all queries on its expected entropy reduction are additive. This assumption allows us to derive a fast greedy approximation of (7), which proves to be effective in empirical evaluations.

Type Bias Correction. PCSs of type t_1 are more likely to produce overlapping query sets than PCSs of type t_2 when $|V_{t_1}| \ll |V_{t_2}|$. In this scenario, a type t_1 node will on average receive more votes than a type t_2 node simply because type t_1 PCSs have a much smaller candidate pool. We correct for *type bias* by discounting nodes with type t by $s_t = 1 - \frac{b}{|V_t \cap \mathcal{U}|+1}$ where $\frac{b}{|V_t \cap \mathcal{U}|}$ is the probability that a node is included in a size b random sample. We add 1 to the denominator for smoothness.

Label Diversity. We enforce label coverage in the query set by promoting PCSs that have suggested queries with diverse labels. Let $\mathcal{L}^\lambda$ be the set of labeled nodes nominated by λ. *Label diversity* of λ is the entropy in $\mathcal{L}^\lambda$, $H(\mathcal{Y}|\lambda) = -\sum_{y \in \mathcal{Y}} P(y|\mathcal{L}^\lambda) \log P(y|\mathcal{L}^\lambda)$, highest when $\mathcal{L}^\lambda$ contains an equal number of nodes with each label. This scheme is preferable to distance based coverage enforcement since it does not assume assortativity. Without assortativity, nodes with dissimilar labels are not necessarily far apart, which implies requiring a minimum distance between queries does not lead to guaranteed coverage. Furthermore, $H(\mathcal{Y}|\lambda)$ is much cheaper to compute than a set with a minimum distance.

IV. Empirical Evaluation

We evaluate MABAL on three classification tasks over two real world datasets against simple heuristic and literature active learning baselines. To demonstrate that MABAL is not dependent on any particular collective classification method, we measure its gain on two different network classification models, RankClass [11] (classification on HINs) and Label Propagation [23] (handles all nodes equally). Both produce probability distributions for label predictions, compatible with our entropy reduction framework. Note that centrality values only need to be computed once for each network and can be shared across classification tasks.

A. Experiment Setup

Datasets. We evaluate our algorithm on HINs constructed from *DBLP*[1], a bibliographic database of computer science publications with 14K papers, 20 conferences, 14K authors and 9K terms, with a total of 171K links, and the MovieLens database consisting of 3.4K movies, 15K crewmen, 2K users, 2.7K tags, and 42 origins, with a total of 435K links [10]. The DBLP HIN contains node types *author, paper, venue*, and *term*, connected via three relations: (paper, author), (paper, venue), (paper, term). The MovieLens HIN contains nodes of type *movie, crew, origin, tag, user*, with relations (movie, crew), (movie, origin), (movie, tag), and (user, movie).

Baselines. We compare against the following baselines: **Random**: b nodes are randomly selected from $\mathcal{U}$. **Single centrality**: nodes in V are queried in descending order by a single centrality measure (degree, PageRank, eigenvector, Katz, betweenness, closeness). **ALFNET** [2]: a cluster-based active learning algorithm on *homogeneous* networks with node features. **MI** [13]: an active learning algorithm on *homogeneous* networks. **HINAL** [21]: an active learning algorithm on heterogeneous networks, which first performs clustering using *metapaths* and then selects queries within each cluster based on uncertainty sampling.

[1] http://dblp.uni-trier.de/

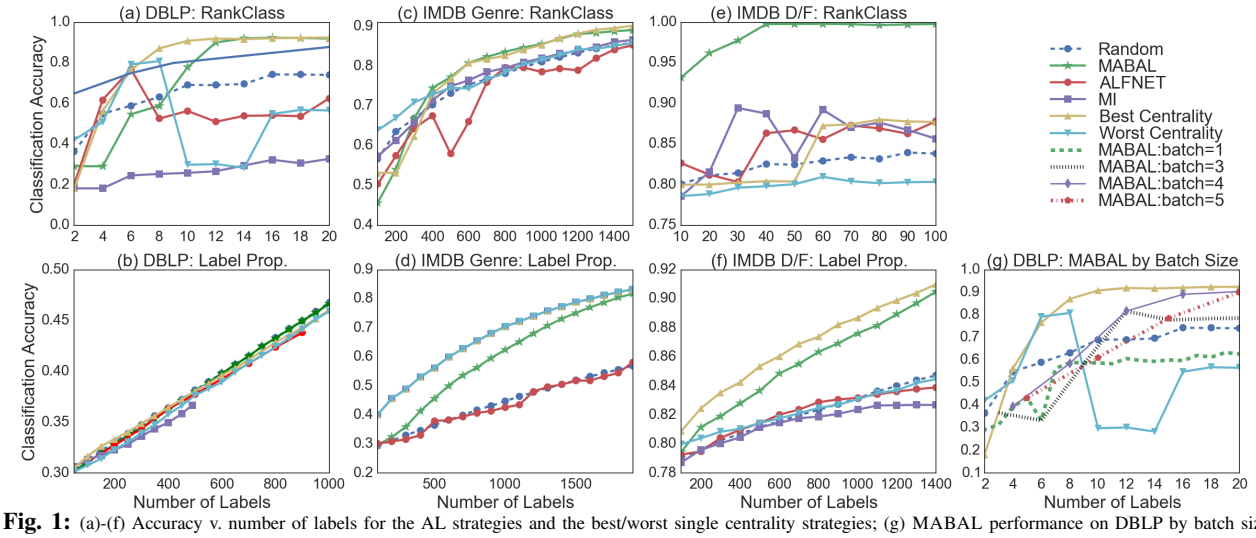

Fig. 1: (a)-(f) Accuracy v. number of labels for the AL strategies and the best/worst single centrality strategies; (g) MABAL performance on DBLP by batch size.

Tasks. For the DBLP network, we classify author, paper, and venue nodes into four research area: {*Data Mining, Database, Machine Learning, Artificial Intelligence*}, with ground truth labels obtained in the same fashion as in [11]. We consider two classification tasks over the MovieLens network: 1) genre classification of movies and actors, 2) domestic vs. foreign classification for movies and actors. The network in our experiments contains 2:1 domestic vs. foreign films and three genres: {*Action, Romance, Thriller*}, with each node belonging to a single genre to satisfy the single label condition in MABAL.
Setup. For fairness of comparison, we perform classification using the same models, RankClass and Label Propagation, on all active learning strategies in the experiments. For ALFNET and MI, which are designed for homogeneous networks, we use the same network topology for input to avoid loss in structural information due to any projection. ALFNET is provided the node type information via node feature vectors.

B. Results

Figure 1 shows the classification accuracy achieved by the number of input labels in each AL algorithm for the three classification tasks. For readability, we include only the best and worse centrality-based strategies to provide reference on the range of performance for MABAL. Note that the centrality baselines are not exactly the primary learners used in MABAL, since each learner only contains a single type of nodes. Due to this fact, MABAL may not fully converge to the best centrality baseline for some tasks, as is the case in Figure 1(d). Note that MABAL achieves competitive performance with the best single centrality baseline even if the worst centrality underperforms Random, as is the case in Figure 1(a), (c), and (d). On the other hand, it also leads to MABAL outperforming all centrality baselines in tasks that contains one node type that is much more informative than the others, as in Figure 1(c).

1) No Consistently Best Baseline: The literature and heuristic baselines alternate in being competitive with MABAL for the four experiment setups. ALFNET and MI are significantly outpaced by MABAL due to their inability to properly handle relation semantics in HINs. HINAL, the only baseline designed for HINs, achieves competitive performance with MABAL on the DBLP dataset, but we were unable to run it on the other tasks. Figure1(e) shows that there is not a single best centrality measure for all tasks, although betweenness and PageRank tend to outperform the others in most cases. Betweenness and closeness, unlike the other centralities, scale super-linearly with the number of nodes. On DBLP and IMDB, they are two orders of magnitudes more time consuming to compute than the others. Thus, it is worthwhile to compute the other centralities and to use MABAL to detect when betweenness fails for a given task. Additionally, informativeness of node types is much more ad hoc and difficult to determine *a priori*. Figure 1(c) shows that MABAL is able to automatically take advantage of informative node types when applicable.

2) Effect of Batch Size: As seen in Figure 1(g), batch mode with $b > 1$ provides significant gain over fully sequential learning, i.e., $b = 1$, because it avoids being pigeonholed into a subgraph and losing coverage. While a smaller batch size leads to faster convergence to λ^{OPT}, it also requires more frequent retraining. The choice for batch size thus involves consideration for the tradeoff between label cost and the cost of model training.

V. Related Work

One important type of analyses on networks is *collective classification*, which accounts for data dependencies when classifying objects in a network [15]. There are two main approaches to collective classification: (1) utilizing local conditional classifiers and (2) viewing it as a global objective optimization problem. In our work, we employ two methods, Label Propagation [23], from the second category, and RankClass [11], a hybrid.

The expected error reduction framework we use was first proposed for text classification [14]. It was then adapted to graphs [24]. However, their formulation relies on the assumption that nodes with the same labels are in close proximity to each other, a limitation excluded from our framework.

Previous works on network active learning without the assortativity assumption do not have a mechanism to make use

of the node type information in HINs [13], [19]. Additionally, [19] requires a pairwise similarity matrix as input, which may not be available for some problem settings, such as HINs where nodes of different types are not comparable. [13] runs in exponential time for some settings, while our algorithm runs in polynomial time. The mutual information query selection criterion in [13] is similar in spirit to our centrality-based primary strategies. The network active learning framework presented in [2] employs both a local classifier based on node attributes and a collective classifier to account for data dependencies. It relies on clustering on node attributes to avoid sampling bias, which is ineffective for spares node attributes. In comparison, we avoid bias by directly using the observed label distributions in the primary strategies.

To the best of our knowledge, [21] is the only existing work on active learning on HINs. In their study, a combination of clustering using metapaths and uncertainty sampling is used for query selection. We use an expected error reduction scheme instead of clustering in our work because we have found that computing clusters that correlate well with class labels in an HIN is highly sensitive to the relations considered. Instead of depending on user guidance in the form of metapaths for performance, we provide to the users information about the network structures that are crucial to their task.

Active search is a class of problems closely related to AL, where the goal is to discover, under a query budget, as many instances of a given class as possible for binary classification [8]. This is applied to graphs in [22], solving a binary version of our problem. Extending this to handle > 2 classes is nontrivial since the algorithm needs to coordinate among a set of binary learners competing for the same query budget.

As a resource allocation model, multi-armed bandit lends itself naturally to the active learning problem. Prior works have independently drawn analogies between active learning and multi-armed bandit [1], [7]. They explore fully sequential learning, i.e., a single query is made at every iteration. As seen in our work, the correspondence between AL and MAB becomes much more complex in batch mode learning. To this end, we transform the batch mode active learning problem in order to apply a combinatorial MAB algorithm [4].

VI. Conclusion

We presented a novel and effective active learning algorithm for heterogeneous information networks. We focused on batch mode learning, which we have shown to be more effective on information networks than fully sequential learning. By establishing a correspondence between batch mode active learning on information networks and combinatorial multi-armed bandit, we proposed an expected error reduction based algorithm that combines simple strategies called *primary learners* to form query sets. Our algorithm employs a novel error expectation measure on networks that is highly adaptable to different classification tasks. Results for classification tasks on real world HINs demonstrate that our algorithm outperforms existing methods when applied to both homogeneous and heterogeneous network classification models.

Acknowledgments. Research was sponsored in part by U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), DARPA under Agreement No. W911NF-17-C-0099, National Science Foundation IIS 16-18481, IIS 17-04532, and IIS-17-41317, DTRA HDTRA11810026, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov). Any opinions, findings, and conclusions or recommendations expressed in this document are those of the author(s) and should not be interpreted as the views of any U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] Y. Baram, R. El-Yaniv, and K. Luz. Online choice of active learning algorithms. *JMLR*, 2004.
[2] M. Bilgic, L. Mihalkova, and L. Getoor. Active learning for networked data. In *ICML*, 2010.
[3] U.k Brandes and T. Erlebach. *Network analysis: methodological foundations*, volume 3418. Springer Science & Business Media, 2005.
[4] W. Chen, Y. Wang, and Y. Yuan. Combinatorial multi-armed bandit: General framework and applications. In *ICML*, 2013.
[5] Y. Chen and A. Krause. Near-optimal batch mode active learning and adaptive submodular optimization. In *ICML*, 2013.
[6] D. Easley and J. Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.
[7] R. Ganti and A. G. Gray. Building bridges: Viewing active learning from the multi-armed bandit lens. In *UAI*, 2013.
[8] R. Garnett, Y. Krishnamurthy, X. Xiong, J. Schneider, and R. Mann. Bayesian optimal active search and surveying. *arXiv:1206.6406*, 2012.
[9] J. Gittins, K. Glazebrook, and R. Weber. *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.
[10] M. Harper and J. Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2015.
[11] M. Ji, J. Han, and M. Danilevsky. Ranking-based classification of heterogeneous information networks. In *SIGKDD*, 2011.
[12] A. Kapoor, E. Horvitz, and S. Basu. Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *IJCAI*, 2007.
[13] C. Moore, X. Yan, Y. Zhu, J. Rouquier, and T. Lane. Active learning for node classification in assortative and disassortative networks. In *SIGKDD*, 2011.
[14] N. Roy and A. McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML*, 2001.
[15] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. *AI magazine*, 29, 2008.
[16] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52, 2010.
[17] B. Settles, M. Craven, and L. Friedland. Active learning with real annotation costs. In *NIPS workshop on cost-sensitive learning*, 2008.
[18] Y. Sun and J. Han. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3, 2012.
[19] W. Tong and R. Jin. Semi-supervised learning by mixed label propagation. In *National Conference on Artificial Intelligence*, 2007.
[20] S. Vijayanarasimhan and K. Grauman. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *CVPR*, 2009.
[21] C. Wan, X. Li, B. Kao, X. Yu, Q. Gu, D. Cheung, and J. Han. Classification with active learning and meta-paths in heterogeneous information networks. In *CIKM*, 2015.
[22] X. Wang, R. Garnett, and J. Schneider. Active search on graphs. In *SIGKDD*, 2013.
[23] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. In *Tech. Rep., Technical Report CMU-CALD-02–107, Carnegie Mellon University*, 2002.
[24] X. Zhu, J. Lafferty, and Z. Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, volume 3, 2003.