Active Learning with Partially Featured Data

Seungwhan Moon
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Ave. Pittsburgh, PA 15213, USA
me@shanemoon.com

Calvin McCarter
Machine Learning Department
Carnegie Mellon University
5000 Forbes Ave. Pittsburgh, PA 15213, USA
cmccarte@cs.cmu.edu

Yu-Hsin Kuo
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Ave. Pittsburgh, PA 15213, USA
yuhsink@cs.cmu.edu

ABSTRACT

In this paper, we propose a new active learning algorithm in which the learner chooses the samples to be queried from the unlabeled data points whose attributes are only partially observed. In addition, we propose a cost-driven decision framework where the learner chooses to query either the labels or the missing attributes. This problem statement addresses a common constraint when building large datasets and applying active learning techniques on them, where some of the attributes (including the labels) are significantly harder or more costly to acquire per data point. We take a novel approach to this problem, first by building an imputation model that maps from the partially featured data to the fully featured dimension, and then performing active learning on the projected input space combined with the estimated confidence of inference. We discuss that our approach is flexible and can work with graph mining tasks as well as conventional semi-supervised learning problems. The results suggest that the proposed algorithm facilitates more cost-efficient annotation than the baselines.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*concept learning, knowledge acquisition*; H.2.8 [**Database Management**]: Database applications—*data mining*

General Terms

Algorithms, Experimentation

Keywords

Active learning, cost optimization, graph categorization, partially featured data

1. INTRODUCTION

The challenge in many of the machine learning tasks is that while unlabeled data is abundant, acquiring class labels is costly because it often involves human annotators, lab experiments, etc. The active learning paradigm addresses the challenge of insufficient labels by optimizing the selection of queries. Several studies show that active learning greatly reduces the labeling efforts in various domains, including graph classification [11], network analysis [4], text mining [17], and many other machine learning or data mining tasks.

The underlying assumption of active learning scenarios is that all features of unlabeled data are free or inexpensive to acquire. As such, active learning techniques seek to find the unlabeled samples that would best improve the system performance once they are labeled. A number of query strategy frameworks have been developed by researchers, one of which is uncertainty-sampling [12, 13]. For multi-classification tasks (which we mainly focus on in this paper), the uncertainty-sampling method queries the instance that is the least confident:

$$x^*_{LC} = \underset{x}{\operatorname{argmax}}\ 1 - P_\theta(\hat{y}|x),$$

where x^*_{LC} is the sample to be queried for labels, and $\hat{y} = \operatorname{argmax}_y P_\theta(y|x)$, or the label that has the highest posterior probability under the model θ. One popular improvement on this base sampling strategy is to consider the density around the sample ([19], [16]), penalizing outlier samples that are not representative. Note that these sampling strategies assume that the learner has access to the fully featured unlabeled dataset from which it can make a query.

However, an interesting problem arises when acquiring some of the features becomes costly. For example, consider that we would like to build a large dataset from scratch to classify wines. While some of the primitive attributes such as their weights, colors, etc. are easier to acquire, other physicochemical attributes such as chemical reaction time can only be obtained via actual experiments, which is more costly [5].

Another example is a graph categorization problem on big graph data. Among various ways to acquire features for graph categorization such as graph kernel methods [8, 1], a new method to transform graphs into feature vectors by their topological and label attributes has been proposed [14]. However, computing some of the topological attributes for the entire graph has high computational overhead, which makes it difficult to obtain a fully featured dataset for an extremely large graph, given the limited computing resources. In addition, for a graph categorization task on biological networks, features of a given network are often not computable without further experiments. Thus, given the bud-

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2580062.

get constraint, only a small subset of the dataset can be fully featured, whereas the majority of the data points are only partially featured with free attributes. If we were to use the traditional active learning techniques, the learner is forced to choose the samples from a pool that is only partially observed, which may not be rich enough to learn a true classifier in the output space.

Some of the works on active feature-value acquisition ([18], [15]) also address the costly attributes problem, but most of the works focus on querying for the most informative attributes rather than the labels. In other words, the purpose of active feature-value acquisition is to tune the current system model by optimally choosing the instances to recover the missing features. In this approach, the expected utility is typically obtained by comparing the model's prediction with the true labels that are assumed to be given.

In this paper, however, we present a new approach to learn from the partially observed dataset by first building an imputation model for missing features, and then by performing active learning in the projected input space. We also propose a novel confidence metric for the inference step which is incorporated into the utility function for active selection. We build the imputation model by using the subset of samples that are fully featured. Therefore, our method can be applied even when there are initially no labels available, because the inference model is built within the input space only. We also present a cost-driven decision framework to selectively query for missing attributes or labels when there is a separate cost for them, depending on their maximum utility.

The rest of the paper is organized as follows: Section 2 describes in detail the proposed active learning framework. We discuss the implications of this research in graph mining tasks in Section 3, and we report and analyze the empirical results in Section 4. We give our concluding remarks and proposed future work in Section 5.

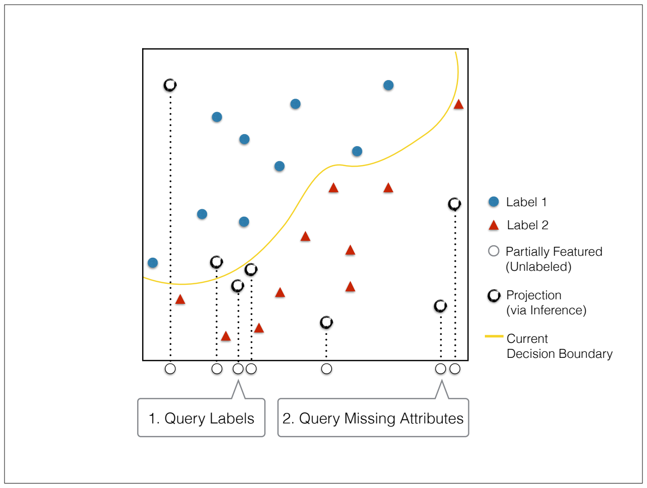
2. METHOD

2.1 Maximum Expected Utility

There are two aspects of optimization in this problem (Figure 1). First, we need to acquire labels that will best improve our current system model (classifier). This is similar to the traditional active learning scenario, except that some of the data points are only partially featured. Second, we may also query for the missing attributes so that we can improve our inference that can map partially featured samples to the fully featured space. In this paper, we assume that there is a separate cost incurred for acquiring labels and for acquiring missing attributes. As such, the ultimate optimization requires an agenda that chooses an instance that would either best improve the system performance or the imputation model at each iteration given the limited budget. In the following subsections, we discuss the two different query types that we can ask to our annotators under the proposed framework.

2.1.1 *Acquiring Labels*

Let us consider N data points $\{x_1, \cdots, x_N\}$ in the input space $\mathbf{X}$, where the corresponding true labels are $\{y_1, \cdots, y_N\}$. The traditional active learning problem can be expressed as finding a solution to the following equation:



● Label 1
▲ Label 2
Partially Featured (Unlabeled)
Projection (via Inference)
Current Decision Boundary
1. Query Labels
2. Query Missing Attributes

Figure 1: Illustration of the proposed algorithm. Given the available labeled data and the current system's decision boundary, the learner may choose between the two different query actions depending on which action has a higher value: (1) it may choose to ask for labels (e.g. of the samples that are near the current decision boundary), or (2) it may ask for filling in the missing attributes (e.g. of the points with low inference confidence).

$$x^* = \underset{x \in \mathbf{X}}{\operatorname{argmax}}\ V(x),$$

where x^* is the sample to be queried for labels, and $V(x)$ is the value of information of the sampled data to the learning algorithm. $V(x)$ can be replaced with any active selection criterion discussed above, e.g. the density weighted uncertainty score proposed in [16].

Now, let us assume that some of the attributes are costly to acquire, and thus some of the data points are only partially featured. Let us denote the fully featured subset of $\mathbf{X}$ as X_f, and the partially featured subset of $\mathbf{X}$ as X_p, where $\mathbf{X} = X_f \cup X_p$. We further assume that each $x_f \in X_f$ is a vector of m attributes, denoted $\{x_f^{(1)}, \cdots, x_f^{(m)}\}$. Similarly, each $x_p \in X_p$ is a vector of l attributes for $l < m$, denoted $\{x_p^{(1)}, \cdots, x_p^{(l)}\}$. Note that every attribute in x_p is also included in x_f for $l < m$.

Then, we train an imputation model $T : X_p \to X_f$ such that it transforms the data from the partially featured subset X_p to the fully featured X_f. For example, for some tasks, T can be a multivariate regression model that fits x_p to x_f, or a variant of EM inference algorithm that predicts x_f from x_p. We present the EM inference model that we used in our paper in Section 2.2. Thus, we can reformulate our task as:

$$x^* = \underset{x \in \{X_f \cup X_p\}}{\operatorname{argmax}}\ V(T(x)),$$

where $T(x)$ is a projection of $x \in X_p$ into the fully featured space, and $x = T(x)$ for $x \in X_f$. Note that this may not return an ideal solution if T is not accurate. In order to account for the confidence level of the imputation model T on a given data point, we denote $\alpha(x_p)$ as the confidence level of T inferring $x_f' = T(x_p)$, where $0 \leq \alpha(x) \leq 1$ for

$x \in X$. Finally, we can write the maximum expected utility $U(x)$ as:

$$x^* = \underset{x \in \{X_f \cup X_p\}}{\operatorname{argmax}} \ \lambda \cdot V_{norm}(T(x)) + (1-\lambda) \cdot \alpha(x), \qquad (1)$$

where $0 \leq \lambda \leq 1$ is a weight factor, and $V_{norm}(\cdot)$ is a normalized value of samples. Note that $\alpha(x_f) = 1$ for $x_f \in X_f$. As such, the optimization problem can be seen as a competence between the projected value of an unlabeled data point and the confidence of inference by the model.

2.1.2 *Acquiring Missing Attributes*

We optimally acquire missing attributes such that it would improve the current inference model the most, which would consequently make the process of acquiring labels discussed above more precise and efficient. In this paper, we present a simple metric for determining the value of an instance with missing features as follows:

$$x_p^* = \underset{x_p \in X_p}{\operatorname{argmax}} \ IV(x_p) = \underset{x_p \in X_p}{\operatorname{argmax}} \ \rho(x_p) \cdot (1 - \alpha(x_p))^{\beta}, \quad (2)$$

where $IV(x_p)$ is the value of potential error reduction in inference of x_p, $\rho(x_p)$ is the density of samples around x_p in the input space, $\alpha(x_p)$ is the confidence of inference by the model, and β is a weight factor determined empirically. As such, the learner would favor the instances with missing features of which the inference confidences are low, and the instances that have many neighbors.

2.1.3 *Joint Optimization Selections*

At each query iteration, we decide whether to query a sample for labels or to query a sample for its missing features. Acquiring missing features would improve the inference model, whereas acquiring labels would result in a more myopic improvement in the system performance. As such, given the limited budget, the query strategies may depend on the learner's agenda and its preference on taking risks in favor of high return [10]. In this paper, we implement a greedy cost-optimization strategy where the learner chooses the best action at each iteration, where the utility for an action is penalized by its cost [7]. The utility function of an unlabeled point x can be formulated as:

$$(x^*, q^*) = \underset{x \in \{X_f \cup X_p\}}{\operatorname{argmax}} \ \max\left(\frac{V(T(x)}{C_l}, \frac{IV(x)}{C_m}\right), \qquad (3)$$

where q^* is a query action type for the corresponding value functions, C_l is the cost for querying the labels, and C_m is the cost for querying the missing attributes. Note that X_p may include the samples that are labeled already. If the labeled samples are queried for their missing attributes, it would adjust the input space and thus refine the current boundary.

2.2 Imputation Model

An imputation model T can be different depending on the types of the attributes of the dataset. In this section, we describe the Expectation Maximization (EM) inference algorithm for the real-valued attributes used in our experiments. As demonstrated in ([9], [6]), EM training can be naturally extended to perform inference over missing values. We chose to use a Gaussian Mixture Model (GMM) to model the joint density over all features. For objects with missing features, we then exploit the simple parametric form of the conditional Gaussian distribution to impute the missing values.

In our experiments a fixed set of feature(s) is missing from some of the training data. However, in this section we present a general framework that can work with varied patterns of missing data. Consider sample $x^i \in \mathbb{R}^d$, for $i \in [1, N]$, where N is the number of observations in the training set. We then partition x^i into vectors x_o^i and x_m^i, corresponding to its observed and missing features. We describe EM algorithm for a GMM with missing features, where K is the number of components.

2.2.1 *Expectation*

For each Gaussian $j \in [1, K]$, we find the posterior probability of x^i belonging to j. Let μ_j the mean of j be split into μ_o and μ_m. Furthermore, we partition Σ_j into Σ_{oo}, Σ_{om}, $\Sigma_{mo} = \Sigma_{om}^{\top}$, and Σ_{mm}. We first calculate the likelihood

$$q_{ij} = \mathcal{N}(x_o^i; \mu_o, \Sigma_{oo})$$

Then, defining π_j to be the prior probability of component j, we compute

$$p_{ij} = \frac{q_{ij}\pi_j}{\sum_{k=1}^{K} q_{ik}\pi_k}$$

2.2.2 *Maximization*

We recompute the parameters for separately each Gaussian j. To do this, we first assume that x^i comes from Gaussian j and consider x_m^i conditioned on x_o^i, which is also a Gaussian distribution.

$$x_m^{i,j} | x_o^{i,j} = \mathcal{N}(\mu_m + \Sigma_{mo}\Sigma_{oo}^{-1}(x_o^i - \mu_o), \Sigma_{mm} - \Sigma_{mo}\Sigma_{oo}^{-1}\Sigma_{om}) \qquad (4)$$

Thus, we impute any missing values x_m^i to the conditional mean

$$\hat{x}_m^{i,j} = \mu_m + \Sigma_{mo}\Sigma_{oo}^{-1}(x_o^i - \mu_o) \qquad (5)$$

We therefore get $\hat{x}^{i,j}$ from combining x_o^i and $x_m^{i,j}$. Now we incorporate the imputed values when updating μ_j and Σ_j.

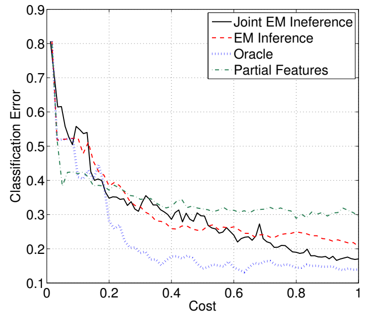
$$\mu_j = \frac{\sum_{i=1}^{N} p_{ij}\hat{x}^{i,j}}{\sum_{i=1}^{N} p_{ij}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{N} p_{ij}(\hat{x}^{i,j} - \mu_j)(\hat{x}^{i,j} - \mu_j)^{\top}}{\sum_{i=1}^{N} p_{ij}}$$

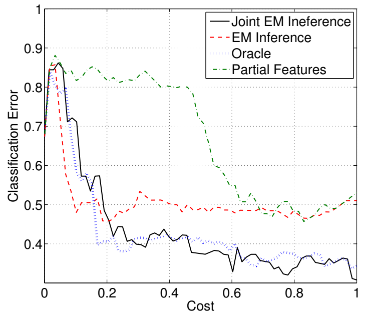
Finally, we recompute π_j by summing the posterior weights.

$$\pi_j = \frac{\sum_{i=1}^{N} p_{ij}}{\sum_{k=1}^{K}\sum_{i=1}^{N} p_{ik}}$$

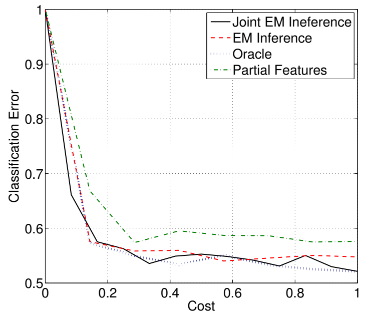
Upon completion of EM training, we impute the missing values without assuming that they come from any particular mixture component. For each x^i with missing features, we draw component j randomly from Multinomial(p_i). We then set $\hat{x}_m^i$ using Equation (5). (Alternatively, we could randomly draw x_m^i using Equation (4).)



(a) Vetebral Column (50% missing 3 attributes out of 6).



(b) Glass (50% missing 3 attributes out of 8)



(c) Wine Quality (50% missing 4 attributes out of 11)

Figure 2: Active learning performance on UCI datasets

2.3 Algorithm

In this section, we present the algorithm that implements our method discussed in Section 2.1 and 2.2. We assume that the data $\mathbf{X}$ is divided into two sets, unlabeled and labeled, which we denote as X_{UL}, and X_L. We denote K as the oracle (annotator), and its answer upon query on $x_{ul} \in X_{UL}$ as y_K. The objective of this task is S, which is the accuracy of the trained classifier. Given the formulated expected utility U (in Equation 1) and the imputation model T, we design our algorithm as follows.

Algorithm 1 Active learning framework

```
function ACTIVE LEARNING($\mathbf{X}$, $T$, $U$)
    $X'_f \leftarrow T(X_p)$        ▷ Apply $T$ on the partailly-featured data $X_p$
    $X_f \leftarrow X_f \cup X'_f$
    while $IterationNotOver$ do
        $(x^*_{ul}, q^*) \leftarrow \underset{x}{\text{argmax}}\ U(x_{ul})$        ▷ From Eq. (3)
        if $q^*$ : query labels then
            $y_k \leftarrow QUERY(x^*_{ul}, K)$
            $X_L \leftarrow X_L \cup \{x^*_{ul}\}$
            $X_{UL} \leftarrow X_{UL} - \{x^*_{ul}\}$
            $Update\ S\ with\ X_L$
        else if $q^*$ : query missing attributes then
            $x_f \leftarrow QUERY(x^*_{ul}, K)$
            $X_f \leftarrow X_f \cup \{x_f\}$
            $X_p \leftarrow X_p - \{x^*_{ul}\}$
            $Update\ T\ with\ X_f$
        end if
    end while
end function
```

3. IMPLICATIONS IN BIG GRAPH DATA

In the previous section, we described the framework for our approach which is flexible and can work with any conventional semi-supervised classification task with continuous features. In this section, we present the implications of the proposed framework in relation to graph mining tasks. A graph categorization problem, for example, is an important and active research topic in graph mining tasks, where the challenge is to efficiently extract features from a graph. However, because most of the featurization techniques involve graph enumeration and graph isomorphism testing which are NP-complete, it is almost impossible to enumerate through the all of the subgraphs and extract features for active learning, especially for large graph data. In addition, extracting features and labels from graphs often requires extensive human efforts, which makes the processing of the dataset for graph categorization more difficult.

[14], for example, proposes a method to extract twenty attributes per node based on its topology and labels. While some of the simple attributes, such as the giant connected component, are easier to compute ($O(n+m)$ where n and m denotes the number of nodes and edges, respectively), there are more complex and informative attributes that require a high computational overhead (e.g. the spectral attributes that depend on eigen decomposition, which has the worst case performance of $O(n^3)$). In addition, features such as label entropy and neighborhood impurity require labeling efforts beforehand, which in some cases are not computable without further annotations or experiments. In contrast, our imputation step to acquire missing attributes only takes $O(nd^2k)$, where d is the number of attributes, k is the number of clusters [20].

The proposed framework thus allows for a new way to tackle the graph mining problem. Using our framework, we propose to mine the graph on a partially-observed basis, and use the imputed features in order to rank the selection queries. This allows us to greatly reduce the efforts in labeling and transforming the graph into features, and to avoid enumerating through the entire graph to extract features. Note that once the method for transforming graphs into feature vectors is determined, the rest of the process can simply follow the algorithm presented in Section 2.

In the following section we present our results on the real datasets using the algorithm discussed above. We choose the typical UCI datasets [2] for multi-classification as representative examples of semi-supervised learning problems. We also present a preliminary result on the Yeast protein interaction network (PIN) dataset.

4. RESULTS AND ANALYSIS

4.1 Baselines

We obtain the baseline performance as follows. We first build a system classifier using only the available labeled data

Table 1: Active learning performance on UCI Vetebral Column Dataset with varying percentage of samples missing some of the attributes (30%, 50%, 70% missing 3 attributes out of 8).

% of Missing Data	Classification Error			
	Cost = 0.25	Cost = 0.50	Cost = 0.75	Cost = 1.00
0 % (Oracle)	0.271	0.201	0.146	0.130
30 %	0.279	0.195	0.182	0.161
50 %	0.347	0.262	0.181	0.164
70 %	0.398	0.303	0.281	0.209
90 %	0.415	0.421	0.413	0.352

Table 2: Active learning improvement on the Yeast Protein Interaction Networks dataset (50% data missing 3 attributes out of 8). Initial classification error rate = 0.701, chance performance = 0.80.

Method	Classification Error			
	Cost = 0.25	Cost = 0.50	Cost = 0.75	Cost = 1.00
Partial Featured	0.641	0.627	0.578	0.574
Oracle	0.569	0.555	0.521	0.463
EM Inference	0.583	0.573	0.541	0.484

points with the reduced-features. In other words, we remove the features corresponding to the missing test feature from the training data. Note that this dataset with missing features might not be rich enough to build a good classifier. We then perform the traditional active learning on this classifier, using the same selection criteria (e.g. entropy-based selection for this experiment) as the proposed algorithm for the value function $V(\cdot)$. We refer this baseline as the *partial features* method. We also present the result when we assume that we have a perfect inference of every missing feature (referred as the *oracle* baseline). We treat this as our soft upper bound.

The two algorithms that we propose are referred as (1) *Joint EM Inference* and (2) *EM Inference*. The *joint EM inference* method allows the learners to choose between the two actions illustrated in Figure 1, depending on the relative values. The *EM inference* method, on the other hand, performs imputation only at the first iteration using the initially available fully featured dataset, and performs traditional active learning with the entropy-based selection criteria.

4.2 UCI Dataset

We test our algorithm on typical UCI datasets for multi-classification task (Vertebral Column (3 classes), Glass (7 classes), and Wine Quality (6 classes)). We assume that some of the features are more expensive to acquire than others for a given dataset, and thus we artificially assign some of the attributes as missing features and hide them from the experiment. For example, the Wine Quality dataset contains 12 physicochemical attributes (e.g. fixed acidity, volatile acidity, citric acid, etc.) of continuous numbers, where acquiring each attribute incurs a varying cost. For this experiment, we assume that the more informative attributes are more expensive, and that some portion of the samples are missing for those attributes. In addition, we assume that querying the labels incurs cost 3 times higher than querying the missing attributes. We ran the experiment 5 times each with a different random allocation of X_f and X_p as well as a different initial labeled set, and we report the average over the runs. Figures 2a, 2b, and 2c show the results when 50% of the data were assumed to be missing from a set number of attributes. X-axis has been normalized by a fixed budget.

In Figures 2a, 2b, and 2c, it can be seen that the proposed algorithms (*EM Inference* and *Joint EM Inference*) perform better than the baseline (*Partial Features*), with $p < 0.01$ on a paired two-sided t-test. The *joint EM inference* method outperforms the *EM inference only* method, which indicates that acquiring the missing attributes incurs a higher information gain even at the cost of acquiring fewer labels. While the *EM inference* algorithm converges at a higher error rate than the oracle method (which assumes a perfect inference model) due to the errors in inference of X_p, the joint method converges at a similar error rate with the oracle method. This indicates that the proposed algorithm can almost reach the oracle performance at a significantly lower cost.

Table 1 shows the different performance of the EM inference algorithm on the Vertebral Column dataset depending on the initial proportion of the samples that are missing some of the attributes. If a higher percentage of the samples is partially featured at the beginning, the system cannot build a confident inference model due to the small number of fully featured data points available to train. This is well represented in Table 1: we observe that with the dataset of which 30% is partially featured, we can achieve almost the same performance as the oracle performance. As expected, the performance degrades as we increase the percentage of the partially featured samples.

4.3 Protein Interaction Networks

We also present a preliminary result of our method on graph datasets. The extension to graph mining tasks is straightforward, because one can cast the graph categorization problem as a semi-supervised learning problem with the numerical features extracted from graphs, as proposed in literature extensively.

We apply our method on the protein interaction network (PIN) of Yeast dataset [3], which consists of 2361 nodes and 7182 edges, where each node is encoded as one of the 13 PIN classes. We use the subset of the dataset with the five most frequent classes. For our experiment, we implement some of the topological attributes proposed by [14] (e.g. clustering coefficient, neighborhood connectivity, eccentricity, average degree, etc.), and assume that 50% of the data are missing 3 out of 8 attributes.

Table 2 shows the active learning performance of three different methods (*EM Inference*, *Oracle*, *Partial Features*) at each interval. It can be seen that the *EM Inference* method outperforms the *Partial Features* baseline at every interval. Note that the *EM Inference* method has a similar performance as the *Oracle* method even with the constraint that the dataset was initially only partially featured. This result implies that applying our methods to graph categorization can result in the reduced costs and efforts in transforming graphs into features, as well as in annotating graphs with labels. This can be regarded as an additional aspect of optimization for building a big graph dataset for mining and learning.

5. CONCLUSION

The proposed algorithm has been shown effective in the selected UCI datasets and the Yeast Protein Interaction Network dataset. Specifically, it was shown that with the dataset of which 30% is partially featured, we can build a classification model that is almost as good as the model built with the fully featured dataset. This implies a practical benefit when building a new dataset: given a limited budget, we may not need to acquire features for all of the data samples that we have, but may choose to leave some portion of the dataset partially featured. As such, the proposed algorithm can save the time and expenses that would normally take to build a fully featured dataset, which is a significant improvement especially for large unlabeled datasets. In addition, the proposed framework can be applied in a variety of tasks, including graph classification, text mining, etc.

There is more work that needs to be done to further develop this research. First of all, we plan on performing more experiments on various types of graphs, and show the efficacy of our method in saving costs and computational overhead for extracting features from graphs. Secondly, we can improve the framework by formulating a long-term agenda for choosing between the two query types, rather than a myopic optimization method. The long-term agenda would outperform the proposed greedy implementation especially when the total budget is variable. Finally, we can generalize the proposed EM algorithm method such that it can be used for other types of datasets as well, for example where the missing attributes are binary or categorical.

Acknowledgements

We would like to acknowledge the contributions made by Professor Seyoung Kim for her valuable and constructive suggestions.

6. REFERENCES

[1] G. Arthur, O. Bousquet, A. Smola, and B. Scholkopf. Measuring statistical dependence with hilbert-schmidt norms. *16th International Conference, ALT*, pages 63–77, 2005.

[2] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[3] V. Batagelj and A. Mrvar. Pajek datasets (yeast), 2006.

[4] M. Bilgic, L. Mihalkova, and L. Getoor. Active learning for networked data. *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[5] P. Cortez, J. Teixeira, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Using data mining for wine quality assessment. *Discvoery Science, J.G. et al., Ed., vol. LNCS 5808.*, pages 66–79, 2009.

[6] O. Delalleau, A. Courville, and Y. Bengio. Efficient em training of gaussian mixtures with missing data. *CoRR*, 2012.

[7] P. Donmez and J. G. Carbonell. Proactive learning: Cost-sensitive active learning with multiple imperfect oracles. *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, 2008.

[8] T. Gartner, P. Flach, and S. Wrobel. On graph kernels: Hardness results and efficient alternatives. *Conference on Learning Theory*, 2003.

[9] Z. Ghahramani and M. I. Jordan. Learning from incomplete data. *Technical Report, MIT*, 1994.

[10] A. Kapoor and R. Greiner. Budgeted learning of bounded active classifiers. *KDD Workshop on Utility-Based Data Mining*, 2005.

[11] X. Kong, W. Fan, and P. S. Yu. Dual active feature and sample selection for graph classification. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2011.

[12] D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. *Proceedings of the International Conference on Machine Learning (ICML) '94*, pages 148–156, 1994.

[13] D. Lewis and W. Gale. Training text classifiers by uncertainty sampling. In *Proceedings of ACM-SIGIR Conference on Information Retrieval*, 1994.

[14] G. Li, M. Semerci, B. Yener, and M. Zaki. Graph classiïňĄcation via topological and label attributes. 2011.

[15] P. Melville, M. Saar-Tsechansky, F. Provost, and R. Mooney. An Expected Utility Approach to Active Feature-Value Acquisition. *Proceedings of the 4th IEEE International Conference on Data Mining*, 2005.

[16] H. T. Nguyen and A. Smeulders. Active learning using pre-clustering. *International Conference on Machine Learning*, 2004.

[17] B. Ribeiro and C. Silva. On text-based mining with active learning and background knowledge using svm. *Soft Computing*, 11, 2007.

[18] M. Saar-Tsechansky, P. Melville, and F. Provost. Active Feature-Value Acquisition. *Management Science*, 2009.

[19] B. Settles and M. Craven. An Analysis of Active Learning Strategies for Sequence Labeling Tasks. pages 1069–1078, 2008.

[20] M.-S. Yang, C.-Y. Lai, and C.-Y. Lin. A robust em clustering algorithm for gaussian mixture models. *Pattern Recognition*, 45(11):3950–3961, 2012.