GRAPH-BASED ACTIVE SEMI-SUPERVISED LEARNING: A NEW PERSPECTIVE FOR RELIEVING MULTI-CLASS ANNOTATION LABOR

Lei Huang, Yang Liu, Xianglong Liu, Xindong Wang, Bo Lang

State Key Laboratory of Software Development Environment, Beihang University, P.R.China
{huanglei, liuyang, xlliu, langbo}@nlsde.buaa.edu.cn

ABSTRACT

Semi-supervised learning and active learning are important techniques to build more accurate model while labeled data are scarce. The objective of this paper is combining both to effectively relieve user labor for multi-class annotation. We propose a novel graph-based active semi-supervised learning framework which aim at efficiently learning a multi-class model with minimal human labor. In particular, we propose Minimize Expected Global Uncertainty algorithm to actively select examples (for labels), which naturally integrates with the probabilistic results of graph-based semi-supervised learning. Meanwhile, we update the model incrementally by decomposed formulation while the new example are incorporated for training, which only has the time complexity of $O(n)$, compared to the original re-training of $O(n^3)$. Extensive evaluations over three real-world datasets demonstrate that our proposed method has the superior performance comparing with the baselines and the capability to efficiently build more accurate model with fractional human labor.

Index Terms— Semi-supervised learning, active learning, multi-class classification, image annotation

1. INTRODUCTION

Given the explosion of the digital images in real world, it is necessary to collect, classify, organize them using an easy, fast and efficient way. Automatic image annotation plays a crucial role in providing feasible solutions via building statistical model, which can significantly reduce the human labor for labeling the images manually. However, to build statistical model, the labeled training examples are essential and indispensable. How to build more accurate model with as few as labeled examples?

Semi-Supervised Learning(SSL) [1] and active learning [2] are potential solutions for this problem. SSL is designed to improve the generalization ability of the supervised learning by the leverage of unlabeled examples. Typical SSL methods include self-training, co-training, transductive SVM, graph-based methods [3, 4], etc. As an important branch of SSL, Graph-Based SSL (GB-SSL) methods have been widely adopted for image or video annotation [5, 6, 7, 8]. Active learning learns a model in an interactive way, which is able to select the most representative data based on the model learned in each iteration. It has been widely explored in multimedia community [9, 10, 11, 12] for its capability of reducing human labor.

Since SSL and active learning are important techniques when labeled data are scarce, how to effectively combining both is meaningful. Song et al [13] proposed an active learning method based on co-training in video annotation. Zhu et al [14] proposed an active learning approach based on a GB-SSL method, in which the reduction of expected risk of labeling each sample can be predicted without retraining the classification model. Jiang et al [7] developed a graph-based SSL method for video concept detection and used active learning to select data-concept pairs for human annotation. Among the previous methods, almost all are in a transductive setting [15] where the model only predicts the given unlabeled training data, and is not required to make predictions outside. Besides, most methods only consider the binary-class problems.

Different from the conventional methods, we propose a novel graph-based active semi-supervised learning framework which can efficiently learn a multi-class model with minimal human labor and work in an inductive setting [15], in which the newly collected unlabeled examples can be predicted and utilized to train the model incrementally. Considering that most GB-SSL methods can naturally provide the probabilistic outputs, which is convenient to measure the uncertainty of all unlabeled examples, we therefore adopt the uncertainty measurement. For a certain unlabeled example, if we incorporate it along with its assumed label (It can be empirically evaluated based on the current model predictor.) and re-train the model, which can make the new predictor have most confidence, then we should select this example to query the user for label. Based on this intuition, we propose an active learning algorithm, namely Minimize Expected Global Uncertainty (MEGU). Besides, we propose an method to update the model incrementally by means of the decomposed formulation and weighted neighbors assignment, which has

This work was supported by the National Natural Science Foundation of China (Grant No. 61370125) and Foundation of the State Key Laboratory of Software Development Environment (Grant No. SKLSDE-2013ZX-05 and No. SKLSDE-2014ZX-07).

the time complexity of $O(n)$, compared to the original re-training of $O(n^3)$.

To summarize, we highlight the main contributions of our work:

- We propose a novel graph-based active semi-supervised learning framework which can efficiently learn a multi-class model with minimal human labor and work in an inductive setting.
- We propose Minimize Expected Global Uncertainty (MEGU) algorithm to actively select example, which naturally combine the probabilistic outputs of GB-SSL methods. Extensive evaluations over three large datasets demonstrate the superior performance.
- We propose an incremental model updating method, which has the time complexity of $O(n)$, compared to the original re-training of $O(n^3)$.

2. GRAPH-BASED ACTIVE SEMI-SUPERVISED LEARNING FRAMEWORK

We propose a novel graph-based active semi-supervised learning framework which is amenable to efficiently learn a multi-class model with minimal human labor. Figure 1 illustrates the overall framework.

Initially, we have a handful of randomly selected labeled examples and abundant unlabeled examples referred to as the offline example pool. Optionally, we also consider the situation that a web crawler can ceaselessly collect unlabeled examples referred to as the online example pool. We adopt GB-SSL to propagate the label information to the offline example pool. Thus, we have an initial classification model. If we want to make model more accurate , we can use the active learning algorithm to select the most informative examples to query the user for labels from the offline example pool or online example pool and incorporate the selected examples for training. Particularly, we propose Minimize Expected Global Uncertainty algorithm to select the examples and update the model incrementally, which will be elaborated on.

2.1. Label propagation

Given data points set $\chi = \{\mathbf{x}_1, ..., \mathbf{x}_l, \mathbf{x}_{l+1}, ..., \mathbf{x}_{l+n}\}$ where $\mathbf{x}_i \in R^d$. The first l points $\chi_L = \{\mathbf{x}_1, ..., \mathbf{x}_l\}$ are labeled $y_i \in \mathcal{L} = \{1, ..., c\}$ and the remaining points $\chi_U = \{\mathbf{x}_{l+1}, ..., \mathbf{x}_{l+n}\}$ are unlabeled. The goal is to predict the label $y_j (l+1 \leq j \leq l+n)$ of the unlabeled points.

Let F denote the set of $(l+n) \times c$ matrices. A matrix $\mathbf{F} = [\mathbf{F}_1^T, ..., \mathbf{F}_{l+n}^T]^T \in F$ is a vectorial function $\mathbf{F} : \chi \rightarrow R^c$ which assigns a vector $\mathbf{F}_i$ to each point $\mathbf{x}_i$. The label matrix $\mathbf{Y} = [\mathbf{Y}_1^T, ..., \mathbf{Y}_{l+n}^T]^T$ is described as $\mathbf{Y} \in R^{(l+n)\times c}$ with $\mathbf{Y}_{ij} = 1$ if $\mathbf{x}_i$ is with label $y_i = j$ and $\mathbf{Y}_{ij} = 0$ otherwise.

GB-SSL mainly involves two main components: graph construction and label propagation. A typical assumption

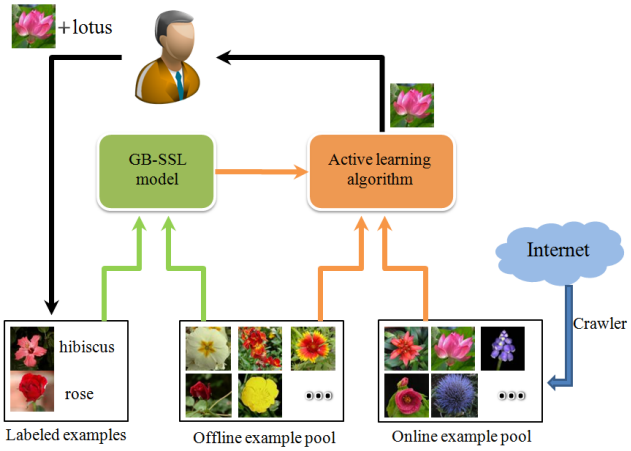


Fig. 1. Illustration of our graph-based active semi-supervised learning framework. Initially, the GB-SSL model is learned with randomly selected labeled examples and offline example pool. Then, we select the most useful examples iteratively from the offline example pool or online example pool for the user labeling and update the model with the new labeled examples.

used in GB-SSL is that nearby points are likely to have the same label. Following Zhou's method [4] with slight difference that we fix the given labels for labeled examples, our label propagation method can be described as follows:

1. Form the affinity matrix $\mathbf{W}$ with its entries $w_{ij} = exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$ if $i \neq j$ and $w_{ii} = 0$.
2. Construct the normalized Laplacian Matrix $\mathbf{S} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$, in which $\mathbf{D}$ is a diagonal matrix with its (i, i)-element equal to the sum of the i-th row of $\mathbf{W}$.
3. Iterate $\mathbf{F}(t+1) = \alpha\mathbf{S}\mathbf{F}(t) + (1-\alpha)\mathbf{Y}$ until convergence, where α is a parameter in (0, 1). Let $\mathbf{F}^*$ denote the limit of the sequence $\mathbf{F}(t)$, which has a closed solution form as :

$$\mathbf{F}^* = \lim_{t\to\infty} \mathbf{F}(t) = (1-\alpha)(\mathbf{I} - \alpha\mathbf{S})^{-1}\mathbf{Y} \qquad (1)$$

4. We can assign each point $\mathbf{x}_i \in \chi_U$ with the label $y_i = \arg\max_{j\leq c} \mathbf{F}^*_{ij}$.

2.2. Inductive setting

Most GB-SSL methods are only for the transductive setting, in which the test examples must be provided before doing the expensive training. For a new test example, it is obligate to execute the algorithm again for predicting the label of the example. The time cost is $O(n^3)$, which is not practicable. Therefor we extend our methods to inductive setting, in which the new test example can be predicted directly by the

learned function $\mathbf{F}$ rather than retaining the model. Following [15]'s method, we fix the graph on $\chi_L \cup \chi_U$ and for a new test point, we propose an induction scheme as follows:

$$\mathbf{F}_x = \frac{\sum_{i \in \chi_L \bigcup \chi_U} w_{xi} \mathbf{F}_i^{nom}}{\sum_{i \in \chi_L \bigcup \chi_U} w_{xi}} \tag{2}$$

where for $i \in \chi_L$, $\mathbf{F}_i^{nom} = \mathbf{Y}_i$. For $i \in \chi_U$, $\mathbf{F}_i^{nom}$ is the normalized value of $\mathbf{F}_i$. Then we assign the test point with label $y_x = arg\max_{j \le c} \mathbf{F}_{xj}$.

2.3. Active learning

By means of the label propagation algorithm, we can get the probabilistic outputs $\mathbf{F}$ (normalizing it if necessary) for all the unlabeled examples (including offline example pool and online example pool). If we want to boost the performance of our model by acquiring another labeled examples, the main issue is how to select most valuable examples to query the oracle (the user) for labels. There exists many sample selection criteria including risk reduction, uncertainty, diversity and so on [11]. For the conventional methods, most criteria are used to measure the unlabeled example(s) itself (themselves) which will be selected to query the oracle. Differently, we consider a global uncertainty. If we add a new example with its assumed label to retrain the model, which can make the new predictor have most confidence over all the unlabeled examples, then we should select this example to query the oracle for label. Based on this intuition, we propose Minimize Expected Global Uncertainty (MEGU) algorithm to actively select examples.

2.3.1. Minimize expected global uncertainty

We formalize our method as follows. Let Ω_U denote the unlabeled examples set (To simplify the notation, we don't distinguish offline example pool and online example pool unless it is necessary.) and Ω_L the corresponding labeled examples. Let $Y_U = \{Y_i\}_{i=1}^n$ be the class membership random variables on Ω_U. $P(Y_U|\Omega_L, \Omega_U)$ is the underlying class conditional probability distributions. Since $P(Y_U|\Omega_L, \Omega_U)$ is unknown, we begin by assuming that we can estimate it with the label propagation algorithm above in Sec 2.1. We hence have:

$$P(Y_U|\Omega_L, \Omega_U) \approx \mathbf{F} \tag{3}$$

We use entropy to measure the uncertainty of a random variable and we assume Y_i are independent. So the global uncertainty can be calculated as:

$$H(\mathbf{F}) = \sum_{i=1}^{n} H(Y_i) = -\sum_{i=1}^{n}\sum_{j=1}^{c} \mathbf{F}_{ij} \log_2 \mathbf{F}_{ij} \tag{4}$$

If we select an unlabeled example $\mathbf{x}_k$ to query the oracle and we receive the assumed label y_k, adding $(\mathbf{x}_k, y_k)$ to the training set and retraining, we will get the new predictor $\mathbf{F}^{+(\mathbf{x}_k, y_k)}$.

Algorithm 1 Minimize Expected Global Uncertainty

```
Input: Ω_L, Ω_U, normalized Laplacian Matrix S;
Initialize F using formula (1);
for each round k do
    for each example x_{k'} ∈ Ω_U do
        for each possible label j ∈ {1, 2, ...c} do
            Compute F^{+(x_{k'}, j)} with Ω_L ∪ {(x_{k'}, j)}
            Compute H(F^{+(x_{k'}, j)}) using formula (5)
        end for
        Compute H(F^{+x_{k'}}) using formula (6)
    end for
    Find x_k based on (7)
    Query x_k for label y_k
    Add (x_k, y_k) to Ω_L, remove x_k from Ω_U
    Update F with the new Ω_L
end for
Output: Ω_L and F .
```

The estimated global uncertainty will also change:

$$H(\mathbf{F}^{+(\mathbf{x}_k, y_k)}) = -\sum_{i=1}^{n}\sum_{j=1}^{c} \mathbf{F}_{ij}^{+(\mathbf{x}_k, y_k)} \log_2 \mathbf{F}_{ij}^{+(\mathbf{x}_k, y_k)} \tag{5}$$

In fact, we don't know the true label y_k before we query the oracle. So we empirically assume the label $y_k = j$ is given with the probability $\mathbf{F}_{kj}$. Hence the expected global uncertainty is:

$$H(\mathbf{F}^{+\mathbf{x}_k}) = \sum_{j=1}^{c} \mathbf{F}_{kj} H(\mathbf{F}^{+(\mathbf{x}_k, j)}) \tag{6}$$

We greedily select the example $\mathbf{x}_k$ that minimizes the expected global uncertainty to query the oracle, which can be formulated as:

$$\mathbf{x}_k = \arg\min_{\mathbf{x}_{k'} \in \Omega_U} H(\mathbf{F}^{+\mathbf{x}_{k'}}) \tag{7}$$

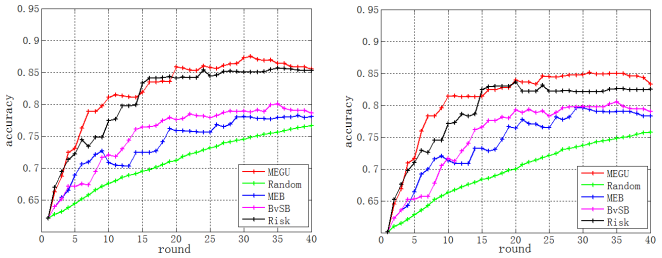
We name it Minimize Expected Global Uncertainty algorithm, which is summarized in Algorithm 1.

2.3.2. Incrementally update

From Algorithm 1, we can find that in each round, we need compute the updated $\mathbf{F}^+$ with $(n \cdot c + 1)$ times where *n* denotes the number of unlabeled examples and *c* denotes the number of classes. If we use formula (1) to compute $\mathbf{F}^+$, which has the time complexity of $O(n^3)$, it is not practicable. We propose an incremental model updating method which reduces the $O(n^3)$ to $O(n)$.

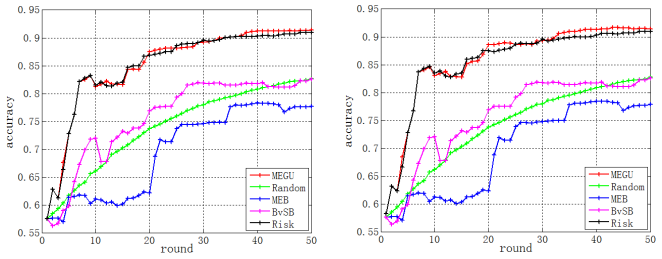
For simplicity, we denote $\mathbf{T} = (1-\alpha)(\mathbf{I} - \alpha\mathbf{S})^{-1}$. If the selected example $\mathbf{x}_k$ is from offline example pool and its given label is $y_k = j$, we can use the decomposed formulation as :

$$\mathbf{F}^+ = \mathbf{T}(\mathbf{Y} + \mathbf{e}_k \cdot \mathbf{e}_j^T) = \mathbf{F} + \mathbf{T}\mathbf{e}_k \cdot \mathbf{e}_j^T \tag{8}$$



(a) transductive accuracy (b) inductive accuracy

Fig. 2. The classification accuracy on USPS dataset. Initially, 10 labeled examples are used to train the model. In each round, one example is selected iteratively for acquiring label and incorporated to retrain the model. (a) shows the transductive accuracy; (b) shows the inductive accuracy.



(a) transductive accuracy (b) inductive accuracy

Fig. 3. The classification accuracy on MNIST handwritten digits dataset. Initially, 10 labeled examples are used to train the model. In each round, one example is selected iteratively for acquiring label and incorporated to retrain the model.

We only need to update the *j*-th column of $\mathbf{F}$ with the increment $\Delta\mathbf{F}_{.j} = \mathbf{T}_{.k}$. which only has the time complexity of $O(n)$.

If the selected example $\mathbf{x}_k$ is from online example pool and its given label is $y_k = j$, we follow [8]'s method with slight changes. The label information of selected example is propagated to its K_{UL} neighbors $\{\mathbf{x}_m\}_{m=1}^{K_{UL}} \subseteq \{\chi_U, \chi_L\}$ with normalized weight w_{km}^{nom}, which can be calculated as follows:

$$w_{km}^{nom} = \frac{exp(-\frac{\|\mathbf{x}_k - \mathbf{x}_m\|^2}{2\sigma^2})}{\sum_{x_m \in N(\mathbf{x}_k)} exp(-\frac{\|\mathbf{x}_k - \mathbf{x}_m\|^2}{2\sigma^2})} \tag{9}$$

According to formula (8), we can update the *j*-th column of $\mathbf{F}$ with the increment as:

$$\Delta\mathbf{F}_{.j} = \sum_{\mathbf{x}_m \in N(\mathbf{x}_k)} w_{km}^{nom} \cdot \mathbf{T}_{.m} \tag{10}$$

where $N(\mathbf{x}_k)$ represents the K_{UL} nearest neighbors of $\mathbf{x}_k$. The computational cost include finding the $N(\mathbf{x}_k)$ and updating $\mathbf{F}_{.j}$ with formula (10), which has the time complexity of $O(K_{UL} \cdot n)$. Generally, K_{UL} is a constant and extremely small. Hence the time complexity is $O(n)$

2.3.3. *Computational cost*

Although we proposed an method to update $\mathbf{F}$ incrementally which dramatically reduce the complexity to $O(n)$, the MEGU algorithm still has the complexity of $O(c^2 \cdot n^2)$ in each round which is not practical for large dataset. For reducing the computational cost further, the possibility strategy is only using the subset of Ω_U. In particular, we only use randomly sampled subset Ω_{sub} with size *m* to select the examples. The trick reduces the $O(c^2 \cdot n^2)$ to $O(c^2 \cdot m \cdot n)$, which is linear to *n*. The subsequent experiment empirically shows that it is practical for real-world scenario.

3. EXPERIMENTS

In this section, experiments were conducted extensively over three real-world datasets to demonstrate the effectiveness of our proposed methods.

3.1. Baselines and evaluation criteria

For comparison, We have implemented Random example selection (Random), Maximize Entropy-Based (MEB) selection [10], Best-versus-Second-Best (BvSB) [10] and Minimize the Risk (Risk) [14][1] as baselines. We partition all the examples into labeled training set χ_L , offline example pool χ_U and online example pool χ_T[2]. We consider both transductive setting and inductive setting. The criteria to compare the performance include the accuracy on offline example pool (transductive accuracy) and the accuracy on online example pool (inductive accuracy). For graph based SSL methods, transductive accuracy measures the propagation effect, and inductive accuracy measures the generalization ability.

3.2. Datasets

The datasets include USPS dataset from the UCI repository[3], MNIST handwritten digits recognition dataset[4] and Flower-102 object categories dataset[5].

USPS: We use the subset of USPS as [3] does. It has 10 classes with unbalanced class sizes. We sampled 3,000 examples as offline example pool and the rest 1,000 examples as online example pool. We randomly sampled 10 examples among offline example pool as initial labeled training set. The digits were preprocessed to reduce the size of each image down to a 16×16 grid, with pixel values ranging from 0 to 255.

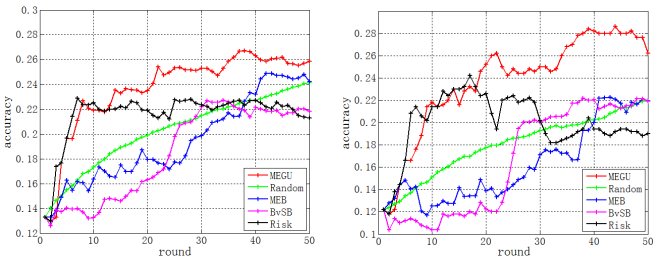
[1]The algorithm described in [14] is only for binary-class classification, and we extend it for multi-class classification.

[2]The main difference between χ_U and χ_T is that χ_U is the unlabeled examples which are used for training the model initially while χ_T is the new test examples.

[3]http://archive.ics.uci.edu/ml/datasets.html.

[4]http://yann.lecun.com/exdb/mnist/.

[5]http://www.robots.ox.ac.uk/ vgg/data/flowers/102/.



(a) transductive accuracy (b) inductive accuracy

Fig. 4. The classification accuracy on Flower-102 dataset. Initially, 10 labeled examples are used to train the model. In each round, one example is selected iteratively for acquiring label and incorporated to retrain the model.

Table 1. Quantitative comparison between MEGU and Random for the number of labeled examples required to achieve certain accuracy on MNIST and USPS dataset (online example pool).

accuracy	dataset	#MEGU	#Random
80%	USPS	19	75
85%	USPS	40	154
80%	MNIST	16	46
85%	MNIST	25	75
90%	MNIST	42	178

MNIST: MNIST is a 10-class handwritten digits dataset, which has a training set of 60,000 examples, and a test set of 10,000 examples. We use randomly sampled 10,000 examples as offline example pool and the official test set as online example pool. We randomly sampled 10 examples among offline example pool as initial labeled training set. Each example is a 28×28 image and is represented by a 784 dimensional vector with values ranging from 0 to 255.

Flower-102: It consists of 102 flower categories. Each class consists of between 40 and 258 images. We randomly selected 12 classes (e.g. rose, lotus, hibiscus, wallflower etc.) to construct a 12-class dataset with 1963 images. We sampled 1463 images as offline example pool and the rest 500 images as online example pool. We also randomly sampled 10 images among offline example pool as initial labeled training set. We extracted dense-sift feature [16], then constructed 1500-D bag-of-words feature via k-means and hard assignment for each image.

3.3. Experiment results

3.3.1. Active learning

In this experiment we compared our proposed MEGU method with the Random, MEB, BvSB and Risk over the three datasets above. For all methods, we use label propagation algorithm described on Section 2.1 to get an initial model, then iteratively select the example for label and incorporate it with the given label to retrain the model. In each round, we evaluate the classification accuracy. For Random, the accuracy was averaged over 50 independent runs. Below are the parameters and the adopted values for each dataset: for USPS dataset, The value of parameter σ is set to 380 empirically [3] and we set $\alpha = 0.89$ after fine tuning. For MNIST and Flower-102, we set $\sigma = 380$, $\alpha = 0.89$ and $\sigma = 21.5$, $\alpha = 0.09$ respectively after fine tuning.

Table 2. Time cost of using subset, compare to the original MEGU and Random methods. Noting that MEGU-*n* indicates the method that using the randomly sampled subset of size *n*.

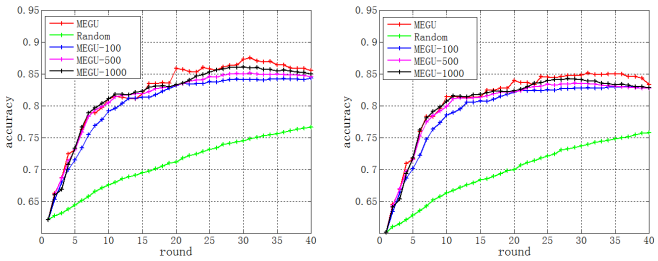
method	time cost (s)
MEGU-100	2.5
MEGU-500	12.8
MEGU-1000	25.7
MEGU	103
Random	0.024

Figure 2, 3 and 4 show classification accuracy on USPS, MNIST and Flower-102 dataset respectively. Given the same size of labeled examples, our proposed MEGU method gives a significantly improved performance compared to Random, MEB and BvSB. Particularly in the initial rounds, MEGU achieved a rapid growth in performance, which demonstrates that MEGU can effectively find the most valuable examples in perspective. Compared to Risk method, MEGU also achieves better performance, especially on Flower-102 dataset. The likely reason is that when calculating the expected global uncertainty or the empirical risk, MEGU considers all possible classes that samples belongs to while Risk only considers the most confident class. Comparing (a) with (b) of Figure 2, 3 and 4, we can find that for all methods, the inductive accuracies and trasductive accuracies are approximately equivalent. Besides, they have a nearly identical trend with the round increases, which strongly suggests that our proposed method for predicting and incorporating newly examples is effective.

Besides, we find that MEB has a worse performance than Random. The likely reason is that for graph-based methods, the samples near the classification boundary are not important, compare to SVM-based methods. Especially when labeled samples are scarce, the samples near the classification boundary may make worse effects due to the label propagation.

3.3.2. Reduction in annotation

In this experiment, we quantify the number of labeled examples required for achieving certain accuracy (e.g. 85%) on online example pool comparing MEGU to Random. The experimental setup is identical to Section 3.3.1. The result can



(a) transductive accuracy (b) inductive accuracy

Fig. 5. Classification accuracy of using subset, compared to the original MEGU and Random methods. MEGU-*n* indicates the method that using the randomly sampled subset of size *n*.

be found in Table 1. We can find that MEGU dramatically reduce the number of labeled examples required for achieving the same accuracy compared with Random. Besides, the results empirically demonstrate the significant advantages of our proposed graph-based active semi-supervised method when require minimal labeled examples to train the accurate model. For example, we only need 42 labeled examples on MNIST training set for building a classification model, which can achieve 90% accuracy for classifying the official test set.

3.3.3. Using subset

In this experiment, we compared the effects of selecting examples only from the subset of the available unlabeled examples. We implemented all methods using MATLAB, running on an 8 GB, 8 core machine with each core at 2.13 GHz. The experimental setup is identical to Section 3.3.1 and the result is evaluated on USPS dataset over 10 independent runs. Figure 5 and Table 2 show the accuracy and time cost respectively. We can find that using subset method has an approximately equal performance compared to the original MEGU, and achieves quite higher accuracy than Random. The time cost is linear to the size of subset. It is worthy noting that MEGU-100 only costs 2.5s while achieves a comparable accuracy than the original MEGU, which is practical for real-world scenario.

4. CONCLUSION

In this paper, we propose a novel graph-based active semi-supervised learning framework which can efficiently learn a multi-class model with minimal human labor (Our method achieved 90% accuracy on MNIST dataset with only 42 labeled examples.). Besides, our proposed MEGU algorithm has superior performance over three real-world datasets comparing with the baselines. Furthermore, we also proposed improved method for efficiency and the comprehensive experiments empirically show that our method is practical for real-world scenario.

5. REFERENCES

[1] Xiaojin Zhu, "Semi-supervised learning literature survey," Tech. Rep. 1530, Wisconsin-Madison, 2005.

[2] Simon Tong and Daphne Koller, "Support vector machine active learning with application sto text classification," in *ICML*, 2000, pp. 999–1006.

[3] X. Zhu, Z. Ghahramani, and J. D. Laffertyz, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003.

[4] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf, "Learning with local and global consistency," in *NIPS*, 2003.

[5] Jinhui Tang, Xian-Sheng Hua, Guo-Jun Qi, Meng Wang, Tao Mei, and Xiuqing Wu, "Structure-sensitive manifold ranking for video concept detection," in *ACM Multimedia*, 2007, pp. 852–861.

[6] Jun Wang, Shih-Fu Chang, Xiaobo Zhou, and S.T.-C. Wong, "Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels," in *CVPR 2008*, 2008, pp. 1–8.

[7] Wei Jiang and Alexander Loui, "Laplacian adaptive context-based svm for video concept detection," in *ACM SIGMM Workshop*, 2011, pp. 15–20.

[8] Lei Huang, Yang Wang, Xianglong Liu, and Bo Lang, "Efficient semi-supervised annotation with proxy-based local consistency propagation," in *ICME*, 2013.

[9] A. Holub, P. Perona, and M.C. Burl, "Entropy-based active learning for object recognition," in *CVPRW*, 2008.

[10] A.J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in *CVPR*, 2009, pp. 2372–2379.

[11] Meng Wang and Xian-Sheng Hua, "Active learning in multimedia annotation and retrieval: A survey," *ACM Trans. Intell. Syst. Technol.*, pp. 10:1–10:21, 2011.

[12] Xin Li and Yuhong Guo, "Adaptive active learning for image classification," in *CVPR*, 2013, pp. 859–866.

[13] Yan Song, Xian-Sheng Hua, Li-Rong Dai, and Meng Wang, "Semi-automatic video annotation based on active learning with multiple complementary predictors," in *MIR*, 2005, pp. 97–104.

[14] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani, "Combining active learning and semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003, pp. 58–65.

[15] Olivier Delalleau, Yoshua Bengio, and Nicolas Le Roux, "Efficient Non-Parametric Function Induction in Semi-Supervised Learning," in *AISTATS*, 2005, pp. 96–103.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005, pp. 886–893.