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 Lab of Software Development Environment, Beihang University, Beijing, China

1. INTRODUCTION

Background

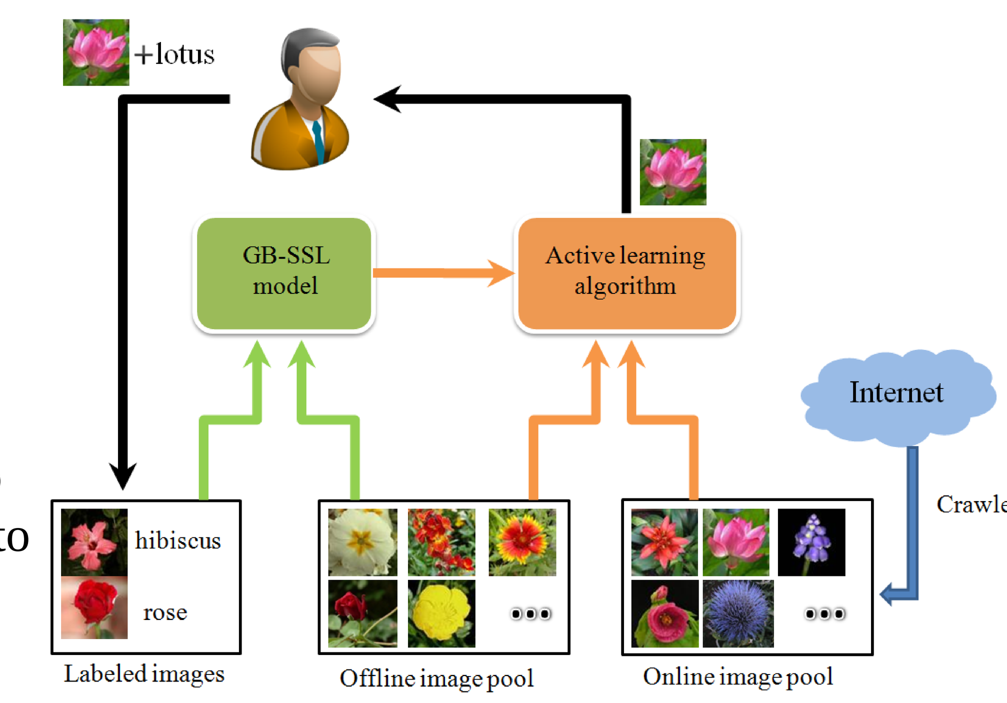
- How to build more accurate model with as few as labeled examples?
- Semi-Supervised Learning + Active Learning
  - Multi-class setting
  - Online data stream
  - Incremental updating

Our Work

- Propose a novel graph-based active semi-supervised learning framework which can learn a multi-class model efficiently with minimal human labor and work in an inductive setting.
- Propose Minimize Expected Global Uncertainty (MEGU) algorithm to actively select example, which naturally combine the probabilistic outputs of GB- SSL methods.
- 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

- **Data Scope**
  - Labeled examples
  - offline example pool
  - online example pool
- **Workflow**
  - Initialize the annotation model by using GB-SSL
  - Use the active learning algorithm to select the most informative examples to query the user
  - Update the model by the selected examples



Label Propagation

- **Notation**
  - A point set $\mathcal{X} = (\mathcal{X}_L, \mathcal{X}_U) = \{\mathbf{x}_1, \dots, \mathbf{x}_l, \mathbf{x}_{l+1}, \dots, \mathbf{x}_n\}$
  - Points $\mathcal{X}_L = \{\mathbf{x}_1, \dots, \mathbf{x}_l\}$ are labeled $y_i \in \mathcal{L} = \{1, \dots, c\}$
  - Predict the label of unlabeled points $\mathcal{X}_U = \{\mathbf{x}_{l+1}, \dots, \mathbf{x}_n\}$
- **Method**
  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 $y_i = \arg\max_{j \leq c} \mathbf{F}^*_{ij}$.

Inductive setting

- **Problem:** For 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)
- **Method:** we fix the graph on $\chi_L \cup \chi_U$ and for a new test point, we propose an induction scheme as follows

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

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 $y_x = arg\max_{j \leq c} \mathbf{F}_{xj}$.

Active learning

- **Notation**
  - Ω_U: the unlabeled examples set
  - Ω_L: the corresponding labeled examples
  - $Y_U = \{Y_i\}_{i=1}^n$: the class membership random variables on Ω_U
  - $P(Y_U|\Omega_L, \Omega_U)$: the underlying class conditional probability distributions , we can estimate it with the label propagation result: $P(Y_U|\Omega_L, \Omega_U) \approx \mathbf{F}$
- **Minimize expected global uncertainty**

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:

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

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 $\mathbf{F}^{+(\mathbf{x}_k, y_k)}$.

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

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:

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

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

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

Algorithm 1 Minimize Expected Global Uncertainty

1: **Input**: Ω_L, Ω_U, normalized Laplacian Matrix $\mathbf{S}$;
2: Initialize $\mathbf{F}$ using formula (1);
3: **for** each round k **do**
4: **for** each example $\mathbf{x}_{k'} \in \Omega_U$ **do**
5: **for** each possible label $j \in \{1, 2, \dots c\}$ **do**
6: Compute $\mathbf{F}^{+(\mathbf{x}_{k'}, j)}$ with $\Omega_L \cup \{(\mathbf{x}_{k'}, j)\}$
7: Compute $H(\mathbf{F}^{+(\mathbf{x}_{k'}, j)})$ using formula (5)
8: **end for**
9: Compute $H(\mathbf{F}^{+\mathbf{x}_{k'}})$ using formula (6)
10: **end for**
11: Find $\mathbf{x}_k$ based on (7)
12: Query $\mathbf{x}_k$ for label y_k
13: Add $(\mathbf{x}_k, y_k)$ to Ω_L, remove $\mathbf{x}_k$ from Ω_U
14: Update $\mathbf{F}$ with the new Ω_L
15: **end for**
16: **Output**: Ω_L and $\mathbf{F}$.

Incrementally update

- **problem**: use formula (1) to update $\mathbf{F}$ or $\mathbf{F}^+$: O(n^3)
- **Incrementally update model with the selected example**
  - New example $\mathbf{x}_k$ from offline example pool with label $y_k = j$
    - ✓ $\mathbf{F}^+ = \mathbf{F} + \mathbf{T}\mathbf{e}_k \cdot \mathbf{e}_j^T$, where $\mathbf{T} = (1-\alpha)(\mathbf{I} - \alpha\mathbf{S})^{-1}$
    - ✓ $\Delta\mathbf{F}_{.j} = \mathbf{T}_{.k}$
  - $\mathbf{x}_k$ from online example pool with label $y_k = j$, propagate the label information to its K_{UL} neighbors with normalized weight
    - ✓ $w_{km}^{nom} = \frac{exp(-\frac{\|\mathbf{x}_k - \mathbf{x}_m\|^2}{2\sigma^2})}{\sum_{x_m \in N(\mathbf{x}_k)} exp(-\frac{\|\mathbf{x}_k - \mathbf{x}_m\|^2}{2\sigma^2})}$
    - ✓ $\Delta\mathbf{F}_{.j} = \sum_{\mathbf{x}_m \in N(\mathbf{x}_k)} w_{km}^{nom} \cdot \mathbf{T}_{.m}$
  - The time complexity is O($K_{UL} \cdot n$)
- **reducing the computational cost further**
  - Using subset: O(c^2 ·n^2) to O(c^2 ·m ·n)
    - ✓ more efficient with compared accuracy

3. EXPERIMENTS

Baselines

- Random example selection (Random)
- Maximize Entropy-Based (MEB)
- Best-versus-Second-Best (BvSB)
- Minimize the Risk (Risk)

evaluation criteria

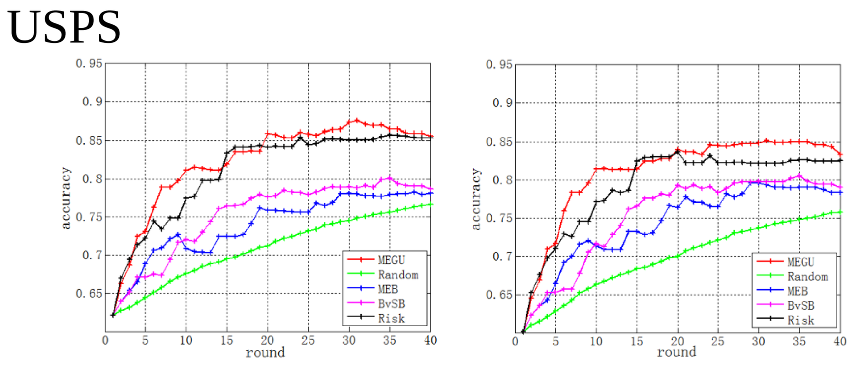
- Transductive accuracy
- Inductive accuracy

Datasets

- USPS : 4000, 10 classes, 256D Pixel
- Flower-102: 1963, 12 classes, 1500D Bow
- MNIST: 70K, 10 classes, 784D Pixel

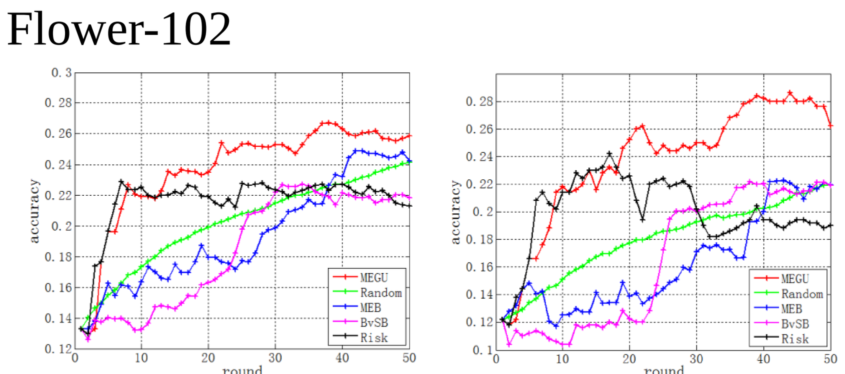
Accuracy comparison

- USPS



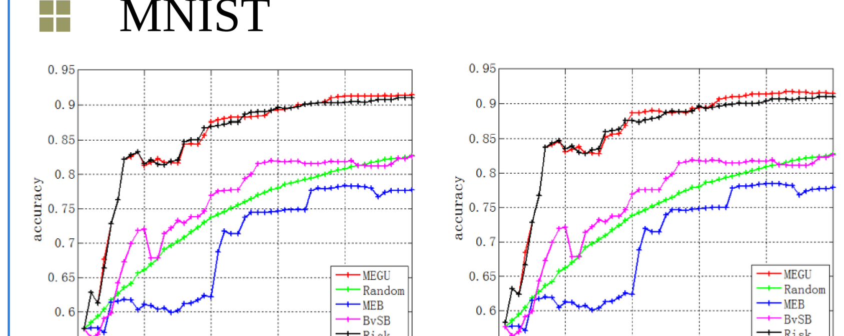
(a) transductive accuracy (b) inductive accuracy

- Flower-102



(a) transductive accuracy (b) inductive accuracy

- MNIST



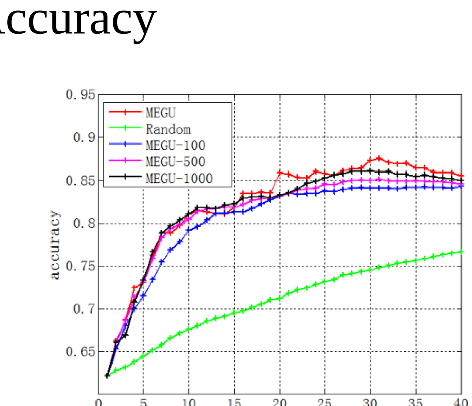
(a) transductive accuracy (b) inductive accuracy

Reduction in annotation

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

Using subset

- Accuracy



- Time cost

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

© July. 2014