Learning to Sample: an Active Learning Framework

Jingyu Shao, Qing Wang and Fangbing Liu
Research School of Computer Science
Australian National University
Acton, ACT, Australia

ICDM-2019

Introduction

■ Motivation

Meta-learning algorithms for active learning are emerging as a promising paradigm for learning the best active learning strategy. However, current learning-based active learning approaches still require sufficient training data so as to generalize meta-learning models for active learning.

AAAI2015-Active learning by learning

NIPS2017-Learning active learning from data

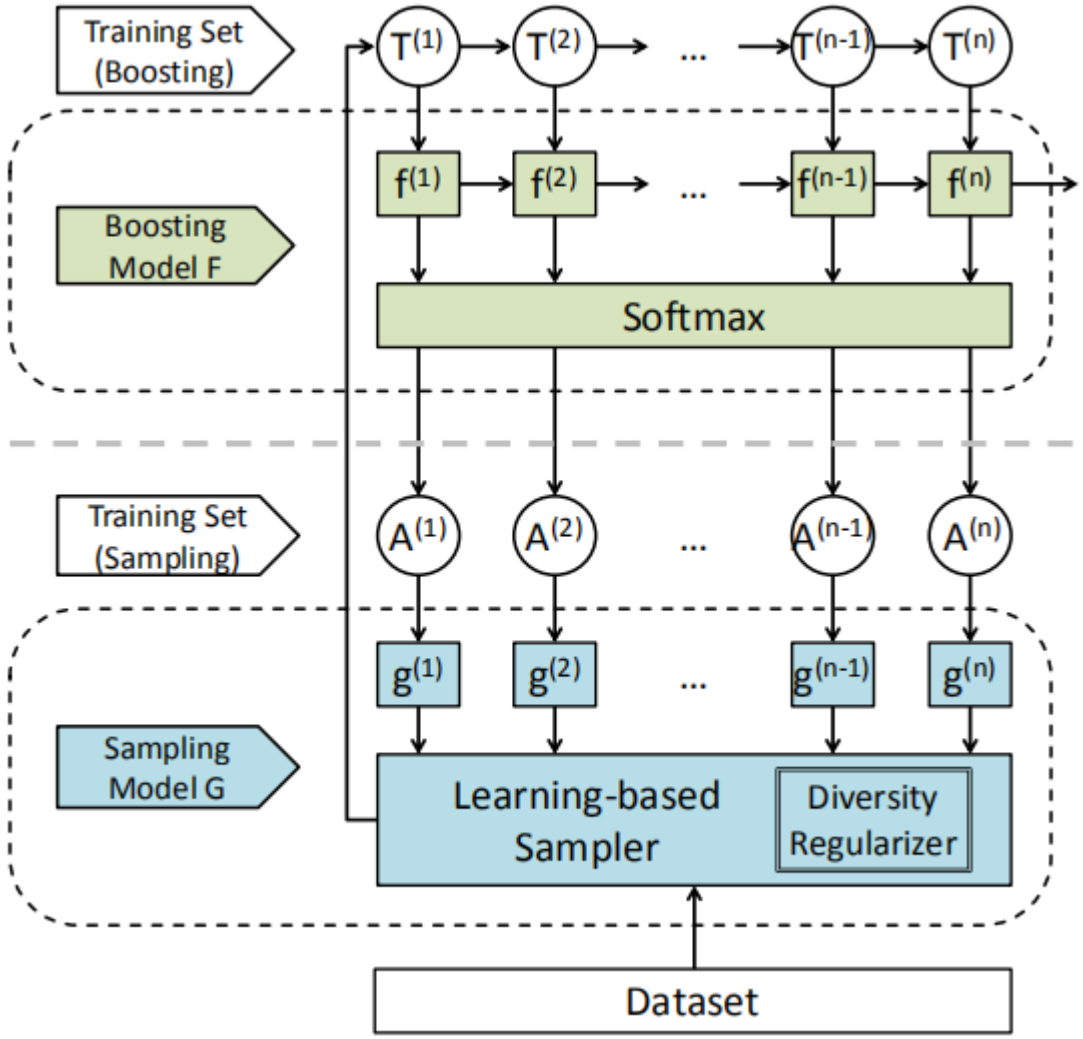
Learning to Sample Framework

■ Framework

In a nutshell, the LTS framework consists of two key components: a **sampling model G** and **a boosting model F**, which are learned iteratively, and their results can mutually strength each other in iterations

F aims to create a strong learner based on a set of weak learners

G is dynamically learned from the performance of the boosting model F during its iterative training process



Learning to Sample Framework

■ Boosting Model

$T = \{(x_i, y_i)\}_{i=1}^{|T|}$, where $x_i \in X$ and $y_i \in R$,

a sequence of training subsets $\langle T^{(1)}, \ldots, T^{(n)} \rangle$ $T^{(1)} \subseteq T^{(2)} \subseteq \cdots \subseteq T^{(n)}$, $T^{(n)} = T$, and $|T^{(n)}| \leq \zeta$

A boosting model **F** trains a sequence of functions $\langle f^{(1)}, \ldots, f^{(n)} \rangle$ in an additive manner,

where $f^{(t)}$ is a function being added into **F** at the t-th iteration.

$$\hat{y}_i^{(t-1)} = \sum_{k=1}^{t-1} f^{(k)}(x_i).$$

$$\sum_{(x_i, y_i) \in T^{(t)}} \ell_1(\hat{y}_i^{(t-1)} + f^{(t)}(x_i), y_i) + \Omega_1(f^{(t)})$$

Learning to Sample Framework

■ Boosting Model

After the t-th function $f^{(t)}$ is learned, the boosting model **F** sends its feedback to the sampling model G via a softmax layer. Model **G** leverage hints from the prediction results of $\langle f^{(1)}, \ldots, f^{(t)} \rangle$and actively select the most informative instances as new samples $T^{(t+1)}$ for the next iteration.

$$\mathbf{l}^{(t)} = \langle \ell(\hat{y}_1^{(t)}, y_1), \ldots, \ell(\hat{y}_q^{(t)}, y_q) \rangle$$

$$z_i^{(t)} = Softmax(l_i^{(t)}), \tag{3}$$

where $Softmax(l_i^{(t)}) = e^{l_i^{(t)}} / \sum_{j=1}^{q} e^{l_j^{(t)}}$ and $l_i^{(t)} = \ell(\hat{y}_i^{(t)}, y_i)$.

Learning to Sample Framework

■ Sampling Model

(1) samples that are likely to be mis-classified by the boosting model uncertainty

(2) samples that have diverse features in the sample space diversity

$$\textbf{maximize} \quad \sum_{i=1}^{k} v_i g^{(t)}(x_i) + \alpha \times \Gamma(\mathbf{v})$$

$$\textbf{subject to} \quad ||\mathbf{v}||_1 = |\Delta^{(t)}|$$

where $k = |X_U^{(t)}|$, $\mathbf{v} = (v_1, ..., v_k)^T \in \{0,1\}^k$ and each v_i is associated with an instance $x_i \in X_U^{(t)}$

The term $g^{(t)}(xi)$ indicates the uncertainty score of an instance x_i which is predicated by a regressor $g^{(t)}$

The regularization term $\Gamma(v)$ controls the distribution of selected instances in order to ensure their diversity in the sample space.

α is a parameter used for balancing the impacts of uncertainty and diversity on samples

Sampling Strategies

■ Uncertainty Sampling

Predict the uncertainty of instances by learning from the performance of the boosting model, i.e. the training loss

a training set $A^{(t)}$ for the sampling model **G** is constructed at the t-th iteration

$$\{(x_i, z_i^{(t)}) | (x_i, y_i) \in T^{(t)}, z_i^{(t)} \in [0,1]\},$$

$$\sum_{(x_i, z_i^{(t)}) \in A^{(t)}} w_i^{(t)} \ell_2(g^{(t)}(x_i), z_i^{(t)}) + \Omega_2(g^{(t)})$$

$w_i^{(t)}$ is a weighted value for x_i and is dynamically adjusted during the iterations.

Sampling Strategies

■ How to decide dynamic weighted values for samples?

Give higher weighted values to samples that are uncertain in more iterations, rather than samples that are uncertain in fewer iterations.

(1) Initialization: For each new sample x_i at the t-th iteration, i.e. a sample in $\Delta^{(t-1)}$:

$$w_i^{(t-1)} = \frac{1}{|\Delta^{(t-1)}|}.$$

(2) Adjustment: Then, the weighted value for each sample xi in $A^{(t)}$ is re-calculated as:

$$w_i^{(t)} = w_i^{(t-1)} \times \frac{e^{-\frac{1}{2} ln(\frac{1-\epsilon^{(t-1)}}{\epsilon^{(t-1)}}) g^{(t-1)}(x_i) z_i^{(t-1)}}}{Z_t},$$

$$\text{where } \epsilon^{(t-1)} = \frac{\sum_i z_i^{(t-1)}}{|T^{(t-1)}|}$$

Sampling Strategies

■ Diversity Sampling

Suppose that unlabeled instances in $X^{(t)}$ are partitioned into a set of groups $\{X_1^{(t)}, \ldots, X_b^{(t)}\}$ alike in certain features. Then we define the regularization term $\Gamma(v)$ over $\{X_1^{(t)}, \ldots, X_b^{(t)}\}$ using a $l_{2,1}$ −norm function as:

$$\Gamma(\mathbf{v}) = ||\mathbf{v}||_{2,1} = \sum_{j=1}^{b} ||\mathbf{v}_j||_2$$

where b is the total number of groups associated with $X_U^{(t)}$, v is partitioned into $\{v_1, \ldots, v_b\}$ where $\Sigma_{j=1}^{b}|vj| = |v|, v_j \in \{0,1\}^m, m = |X_j^{(t)}|$ and $j \in [1, b]$.

$||v_j||_2$ is the $l_{2,1}$−norm of vj that is a binary vector whose elements correspond to instances in group $X_j^{(t)}$

Sampling Strategies

■ How to partition a sample space into groups?

Partition a sample space based on available features of samples.

Given a sample space with d features, a label budget ζ and a number n of iterations, partition the sample space into k groups where $k = \lceil \sqrt[d]{\frac{\zeta}{n}} \rceil^d$ and $\lceil\ \rceil$ indicates the ceiling function.

■ How to distribute label budget across iterations?

Distribute a label budget equally over all iterations, i.e., $|\Delta^{(t)}| = \zeta/n$ for any $t \in [1, n]$

An alternative is to distribute samples in an exponentially decreasing manner over iterations, i.e., $|\Delta(t)| = \zeta/2^t$

Learning To Sample (LTS)

Algorithm 1: Learning To Sample (LTS)

Input: X with k groups, i.e. $\sum_{i=1}^{k} X_i^{(0)} = X$; label budget ζ;
Balancing parameter α; Number of iterations n;

Output: A boosting model F

Initialize $T^{(0)} = \emptyset$

Select a set of seed samples $\Delta^{(0)}$ from k groups to maximize $\Gamma(\mathbf{v})$, where $|\Delta^{(0)}| = \frac{\zeta}{n}$

for $t = 1, \ldots, n$ **do**

  Update $T^{(t)} = T^{(t-1)} + \Delta^{(t-1)}$

  Train an additive function $f^{(t)}$ by minimizing the objective in Eq. 2 using $T^{(t)}$

  Generate a training set $A^{(t)}$

  Train a regression function $g^{(t)}$ by minimizing the objective in Eq. 5 using $A^{(t)}$

  Update $X_i^{(t)} = \{x \in X_i^{(t-1)} | x \notin \Delta^{(t-1)}\}$, where $i = 1, \ldots, k$

  Select a set of samples $\Delta^{(t)}$ from $\sum_{i=1}^{k} X_i^{(t)}$ by maximizing the objective in Eq. 4, with $|\Delta^{(t)}| = \frac{\zeta}{n}$

Experiment

■ Dataset

Classification Tasks	Datasets	# Attributes	# Instances ($\|X\|$)	# Classes	Types of Labels	Class Imbalance Ratio
Image classification	Mnist	28×28	60,000	10	10 digits (i.e. 0-9)	N/A
Salary level prediction	Adult	14	48,842	2	{above 50k, not above 50k}	1 : 3
Entity resolution	Cora	12	837,865	2	{match, non-match}	1 : 49
	DBLP-Scholar	4	168,112,008	2	{match, non-match}	1 : 71,233
	DBLP-ACM	4	6,001,104	2	{match, non-match}	1 : 2,698
	NCVoter	18	10M	2	{match, non-match}	1:420

■ Baseline

CART, Classification And Regression Tree

XG, eXtreme Gradient Boosting

XG+RS, applying XG on training sets built using the random sampling strategy

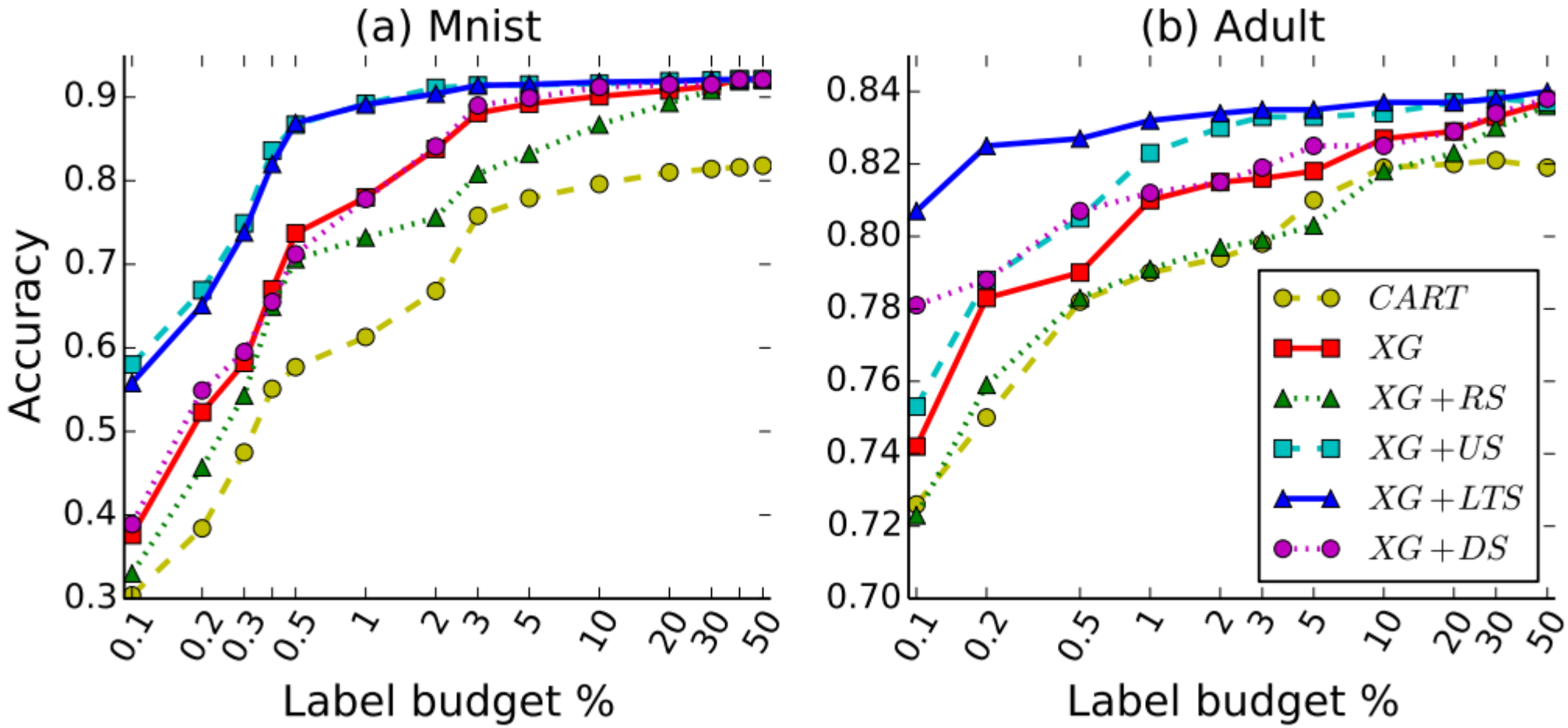
XG+US, applying XG on training sets built only using the uncertainty sampling strategy

XG+DS, refers to applying XG on training sets built only using the diversity sampling strategy

XG+LTS, this paper

Experiment

■ Performance under different label budgets



Experiment

TABLE II: Comparison of f-measure results for entity resolution tasks under different label budgets

Dataset	Label Budget ζ (% of $\|X\|$)	CART	XG	XG+RS	XG + US $\alpha = 0$	XG+LTS $\alpha = 0.5$	XG+LTS $\alpha = 1$	XG+LTS $\alpha = 2$	XG+LTS $\alpha = 5$	XG + DS $\alpha \to \infty$	XG + LTS(E) $\alpha = 1$
Cora	0.01	0	0	0	0	0.637	0.857	0.861	0.867	**0.878**	0.862
	0.05	0.741	0.763	0.750	0.827	0.851	0.864	0.870	0.883	**0.885**	0.867
	0.1	0.788	0.796	0.787	0.823	0.863	0.862	0.873	**0.887**	0.886	0.870
	0.5	0.848	0.835	0.835	0.873	0.893	**0.900**	0.895	0.895	0.893	0.890
	1	0.868	0.878	0.880	0.870	0.896	0.902	**0.904**	0.898	0.894	0.896
	5	0.878	0.897	0.892	0.907	0.912	**0.915**	0.913	0.902	0.898	0.904
NCVoter	0.01	0	0	0	0	0.403	0.324	0.403	0.752	**0.875**	0.571
	0.05	0	0	0	0	0.903	0.954	0.989	**0.993**	0.991	0.934
	0.1	0	0	0	0	0.989	**0.994**	0.993	0.993	0.993	0.993
	0.5	0	0	0	0	0.993	**0.994**	0.993	0.993	0.991	**0.994**
	1	0.334	0.379	0.398	0	0.993	0.993	0.993	0.992	**0.994**	0.993
	5	0.993	0.993	0.994	0.993	0.993	**0.997**	0.993	0.994	0.993	0.994
DBLP-ACM	0.1	0	0	0	0	0	0	0	0	**0.397**	0
	0.5	0	0	0	0	0.382	0.702	**0.720**	0.651	0.632	0.679
	1	0.348	0.347	0.279	0	0.813	**0.878**	0.778	0.730	0.721	0.793
	2	0.599	0.767	0.680	0.403	0.851	**0.884**	0.867	0.789	0.783	0.854
	5	0.870	0.850	0.803	0.874	**0.935**	0.931	0.889	0.837	0.833	0.891
	10	0.903	0.911	0.890	0.926	**0.983**	0.981	0.937	0.893	0.899	0.933
DBLP-Scholar	0.1	0	0	0	0	0.586	0.723	0.733	**0.741**	0.731	0.727
	0.5	0.378	0.54	0.498	0.555	0.764	0.773	**0.794**	0.790	0.780	0.781
	1	0.562	0.669	0.659	0.738	0.793	0.804	**0.808**	0.793	0.792	0.794
	2	0.772	0.806	0.771	0.807	0.810	**0.815**	0.813	0.799	0.801	0.811
	5	0.773	0.822	0.803	0.836	**0.838**	0.836	0.831	0.821	0.818	0.828
	10	0.808	0.835	0.830	**0.865**	0.859	0.851	0.844	0.837	0.829	0.853

Experiment

■ Performance under different sampling distribution methods

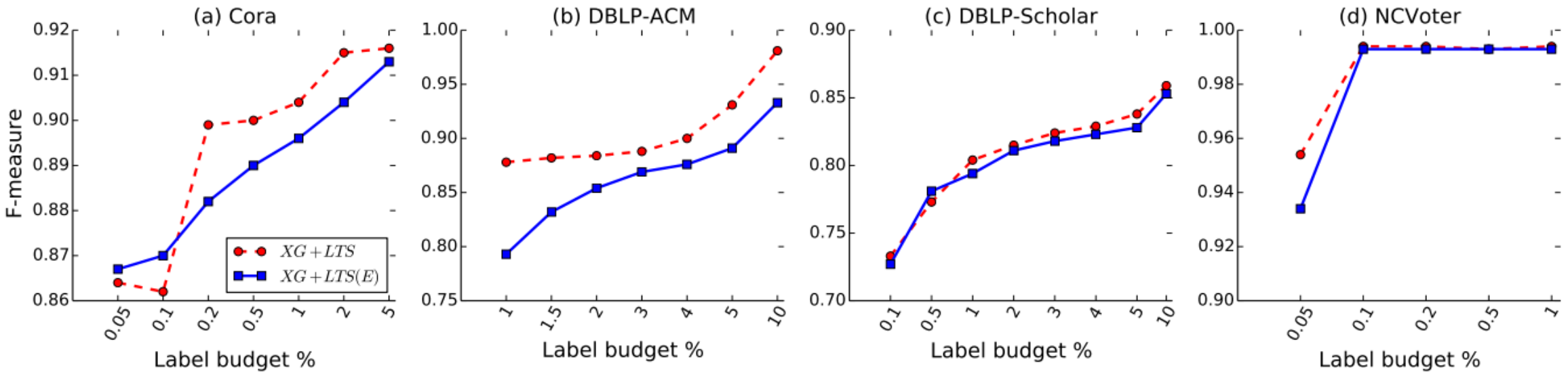
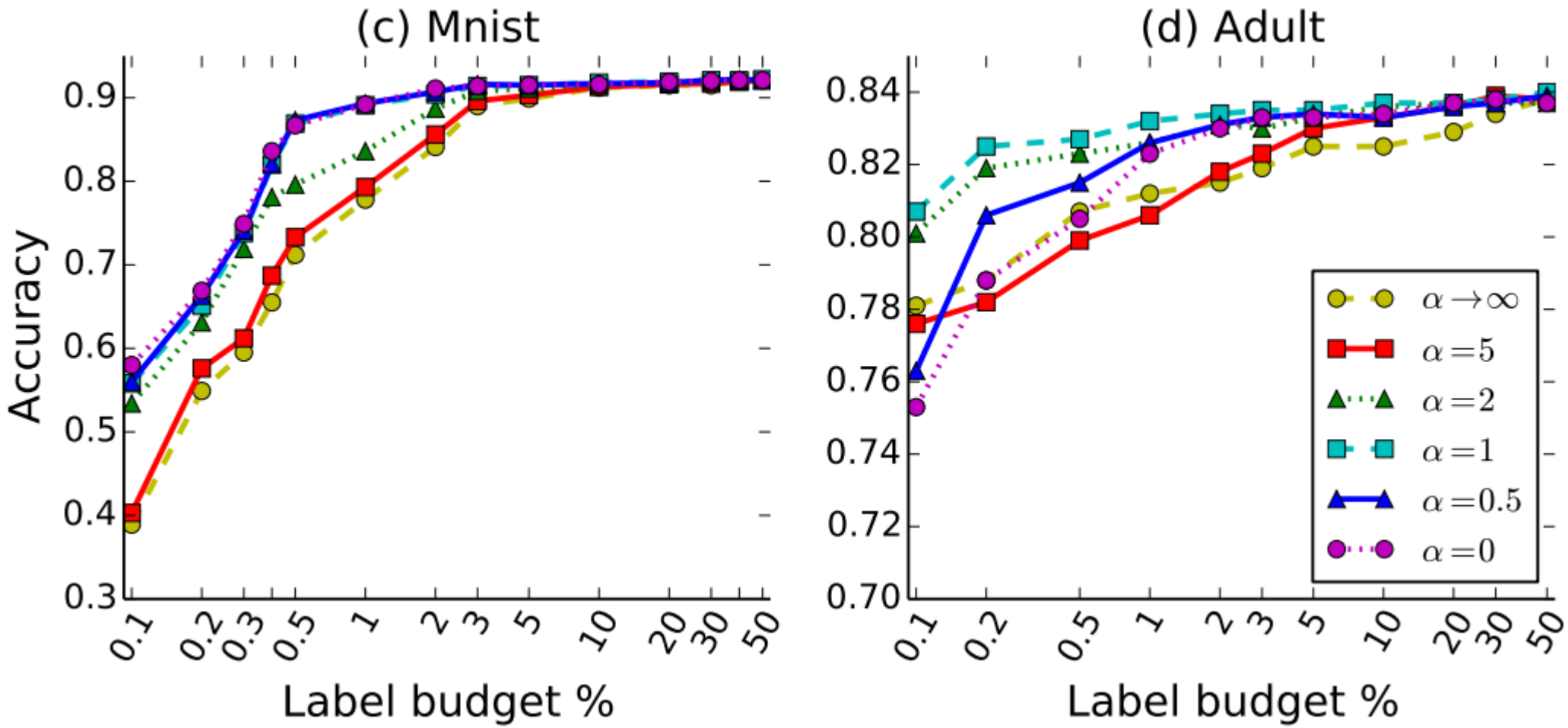


Fig. 5: Comparison of f-measure results for the LTS approach under two different sampling distributions

Experiment

- **Performance under different values of balancing parameter α**



Experiment

- **Comparison of label budgets under the same performance**

TABLE III: Comparison of label budgets w.r.t. classification results with desired FM values, where XG+LTS has $\alpha = 1$.

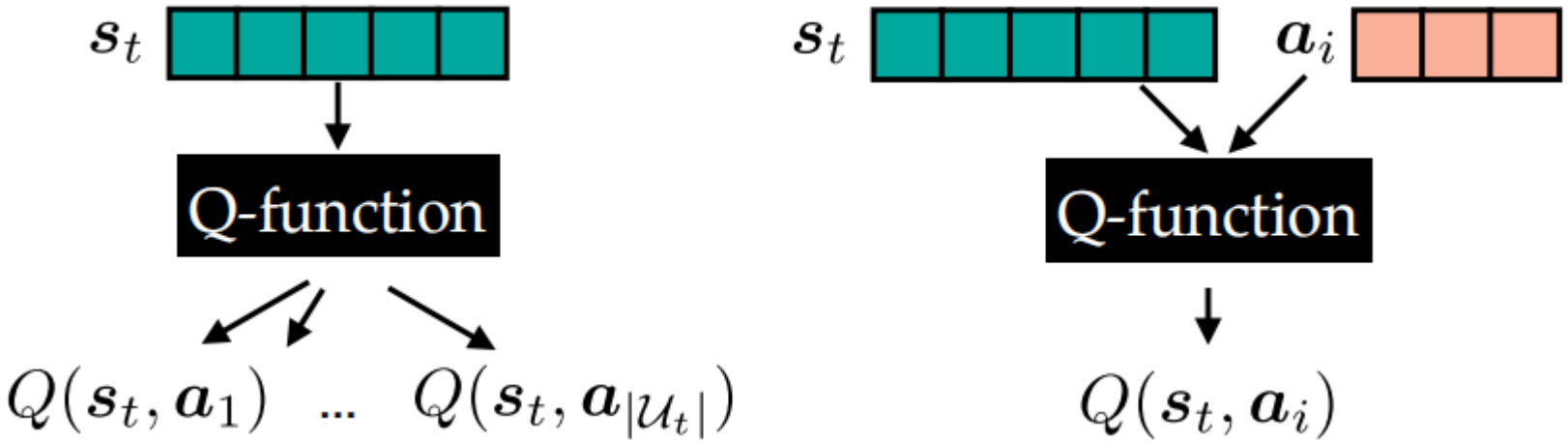
Dataset	Cora	DBLP-ACM	DBLP-Scholar	NCVoter
CART	5%	10%	10%	3%
XG	4%	8%	2%	2%
XG + RS	5%	12%	5%	2%
XG + US	2%	7%	2%	7%
XG + DS	3%	10%	2%	**0.03%**
XG + LTS	**0.5%**	**4%**	**0.9%**	**0.03%**
FM values	0.9	0.9	0.8	0.9

工作汇报

■ 用强化学习的方法学习主动学习采样策略

将学习任务及基模型限定为**图像分类（CIFAR-10）**和**深度残差网络（resnet)**

结合之前meta learning方法，将设计的那些meta特征作为RL的state。按照下列右图，s_t 为设计一些有关当前数据集及模型自身的一些统计特征，a_i为模型针对某一个未标记样本一些特征。



工作汇报

■ 遇到的问题

1.为了获取训练RL的数据，需要在每次query之后重新训练resnet获得reward，重新训练神经网络代价较大耗时很长，即使是用预训练的模型固定除最后一层fc之外layer的参数，训练时间也很长

2.深度模型训练不稳定，实验训练时采用固定epoch的方法。对于epoch大小的选择，实验发现batch-size、epoch大小都会影响模型的收敛。

3.对于深度模型，一个个查询，代价较大，query的形式最好为batch mode。采用batch mode的形式，更新RL的时候就需要进行修改。

工作汇报

■ 拟定的解决方案

$$(s, a, r, s', a')$$

1.训练RL的时候，使用batch mode的方式进行query。此时，reward只能在整个batch查询完后进行模型更新才能获得。目的是为了减少重新训练基模型的次数。

2.Reward的分配，都设置为相同的reward（均分）的；前期reward大，后期reward的小，$reward/2^b$