Bounding Uncertainty for Active Batch Selection

Hanmo Wang,[1,2] **Runwu Zhou,**[1,2] **Yi-Dong Shen**[1*]
[1]State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
{wanghm,zhourw,ydshen}@ios.ac.cn

Abstract

The success of batch mode active learning (BMAL) methods lies in selecting both representative and uncertain samples. Representative samples quickly capture the global structure of the whole dataset, while the uncertain ones refine the decision boundary. There are two principles, namely *the direct approach* and *the screening approach*, to make a trade-off between representativeness and uncertainty. Although widely used in literature, little is known about the relationship between these two principles. In this paper, we discover that the two approaches both have shortcomings in the initial stage of BMAL. To alleviate the shortcomings, we bound the certainty scores of unlabeled samples from below and directly combine this lower-bounded certainty with representativeness in the objective function. Additionally, we show that the two aforementioned approaches are mathematically equivalent to two special cases of our approach. To the best of our knowledge, this is the first work that tries to generalize the direct and screening approaches. The objective function is then solved by super-modularity optimization. Extensive experiments on fifteen datasets indicate that our method has significantly higher classification accuracy on testing data than the latest state-of-the-art BMAL methods, and also scales better even when the size of the unlabeled pool reaches 10^6.

Introduction

Active learning (Settles 2010) is a machine learning and data mining methodology to automatically select informative data instances for annotation when facing a large amount of unlabeled data. The goal of active learning is to train a classifier that has good generalization performance with informative instances only. Traditional active learning methods iteratively select one single informative instance and thus are not efficient when there are multiple annotators. Recently, batch mode active learning (BMAL) was introduced, which makes the annotation process more productive by selecting multiple instances in each iteration.

Informative instances in BMAL are both representative and uncertain. The representativeness indicates that the selected instances capture some global structure of the entire dataset, while the uncertainty indicates that the selected instances can refine the decision boundary in a label-efficient way. Empirically, choosing representative instances are particularly critical when labeled data is scarce, while uncertainty plays a more important role as the number of labeled instances increases (Guo and Schuurmans 2008; Wang and Ye 2015; Settles 2010). There are two common approaches to combine representativeness and (un)certainty. The first is *the direct approach*, which directly combines representativeness with uncertainty to form a single objective, such as in (Wang and Ye 2015), (Chakraborty et al. 2015a) and (Chakraborty et al. 2015b). The second is *the screening approach*, which excludes some unlabeled instances about which the classifier is certain, and chooses representative samples among the remaining instances, such as in (Chattopadhyay et al. 2012) and (Chakraborty et al. 2015b). Neither of these two approaches can handle small number of labeled instances in the beginning of BMAL, when the labeled data is scarce. Under such circumstances, there are not adequate labeled data to train an accurate classifier, and thus the output of the classifier is usually not accurate. The direct approach, which directly utilizes the output, is therefore possibly misled; the screening approach, which screens samples beforehand, may accidentally remove useful instances.

In this paper we propose a novel method that not only alleviates the problems of the direct and the screening approaches but also illustrates the connections between them. In the beginning of BMAL, the output of the classifier is usually not so accurate owing to insufficient labeled data. We observe that samples with high certainty usually remains certain even under a not-well-trained classifier, while samples with low certainty have low confidence on the certainty. To alleviate this issue, we further modify existing certainty measures by enforcing a lower-bound on the certainty score of each unlabeled instance. With this lower-bounded certainty (LBC), some of the most misleading instances have the same certainty score, possibly reducing their influence to the selection process. We then prove that for any representativeness and additive certainty, the direct and screening approaches are two special cases of our LBC method. To the best of our knowledge, this is the first attempt to make generalizations about the direct and screening approaches in literature. Unlike the direct approach, our LBC-based

*Corresponding author

BMAL method treats the noisy samples with equal certainty and thus eliminates some of the most misleading certainty scores. On the other hand, our method takes all unlabeled samples into account, so it may not accidentally discard useful instances merely by their certainty scores as the screening approach does.

As well as the quality of the selected samples, the efficiency of BMAL methods also matters in practice. Unfortunately, most of the existing BMAL methods are not scalable. For example, the most recent method *batchRank* in (Chakraborty et al. 2015b) runs in $\mathcal{O}(n_u^2)$, where n_u is the number of unlabeled instances. The MMD-based BMAL method (Chattopadhyay et al. 2012) and its variant (Wang and Ye 2015) both use a Quadratic Programming toolbox which runs in $\mathcal{O}(n_u^3)$. Another representative BMAL method (Wang et al. 2015) tends to solve its objective function using the sub-gradient method, which has a slow convergence rate. To illustrate the effectiveness, we instantiate our method by choosing Maximum Mean Discrepancy (MMD) (Gretton et al. 2006)(Gretton et al. 2012) as representativeness and posterior probabilities to the most likely class as certainty. Under mild conditions, we prove the super-modularity of our objective function and utilize a random greedy method (Buchbinder et al. 2014) to obtain a fast solution. Our method scales linearly w.r.t. n_u and can handle more than 10^6 unlabeled instances.

Our contributions are as follows:

- We propose to bound the uncertainty of the unlabeled samples for active learning. Such mechanism not only alleviates the problem with the noisy output of the classifier, but also takes all unlabeled samples into consideration. In other words, our method alleviates the problems with the direct approach and the screening approach.
- Our framework can be seen as a generalization of the two aforementioned approaches. To the best of our knowledge, this is the first work for such generalizations. For effectiveness, we empirically choose MMD (Chattopadhyay et al. 2012) and the least confident method as representativeness and uncertainty, respectively. In addition, we discover a trivial solution of the original Quadratic Programming solver in (Chattopadhyay et al. 2012) and propose to solve the objective with a random greedy solver that has theoretical guarantees.
- We conduct extensive experiments on fifteen benchmark datasets. The result demonstrates that our method with LBC has significantly higher accuracy than the latest state-of-the-art BMAL methods, while managing to achieve faster (sometimes in several order of magnitude) running time.

Related Work

Active learning has been an important topic in machine learning and data mining. One extensively studied category is pool-based active learning (Settles 2010), where an active learner is exposed to a large pool of unlabeled data, and it automatically decides which instances are the most informative for labeling. Pool-based active learning methods can be roughly categorized into two groups. One is *single instance selection*, where a single informative instance is selected iteratively to update the classifier. Well-known single instance selection methods include query-by-committee (Seung, Opper, and Sompolinsky 1992) and uncertainty sampling (Tong and Koller 2002). The other category is *multiple instance selection*, a.k.a. batch mode active learning (BMAL), where multiple annotators are available and the learner iteratively selects multiple instances instead of one.

There is a variety of pioneer work in BMAL, which also considers representativeness and uncertainty. For example, (Guo and Schuurmans 2008) proposes to select instances based on pure discriminativeness (which can be seen as a variant of uncertainty). Later, (Guo 2010) proposes an approach to selecting a batch of samples that minimizes the mutual information between labeled and unlabeled data. (Hoi et al. 2006) apply the Fisher information as an uncertainty criterion in BMAL. Recently, (Chattopadhyay et al. 2012) propose a representative method to minimize the difference in distribution between labeled and unlabeled data via selecting the samples with the lowest MMD score. (Wang and Ye 2015) further combine this distribution-matching method with discriminative information. Another method based on the distribution-matching criterion named relative Pearson divergence is proposed in (Wang et al. 2015). (Chakraborty et al. 2015b) present a method using mutual information and entropy. There are also other BMAL methods for specified classifiers such as hierarchical classification (Cheng et al. 2014)(Chakraborty et al. 2015a), logistic regression (Gu, Zhang, and Han 2014), multi-class classifier (Reyes and Ventura 2018; Yan and Huang 2018) and Naive Bayes/Nearest Neighbor (Wei, Iyer, and Bilmes 2015). Recently, there is also theoretical analysis of BMAL(Chen and Krause 2013).

Batch Mode Active Learning

Before diving into our method, we introduce the formal problem setting for BMAL. Let $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ be a dataset of n instances with dimensionality d. We use L and U to denote sets of labeled and unlabeled indexes respectively, where $L \cup U = \{1, 2, ..., n\}$ and $L \cap U = \emptyset$. Let $n_l = |L|$ and $n_u = |U|$. Let X_L and X_U be the labeled and unlabeled set respectively, i.e., $X_L = \{\mathbf{x}_i | i \in L\}$ and $X_U = \{\mathbf{x}_i | i \in U\}$. Each instance $\mathbf{x}_i$ is associated with label $y_i \in \{1, ..., |y|\}$. If $\mathbf{x}_i \in X_L$, label y_i is revealed by a human labeler; otherwise y_i is unknown. A BMAL algorithm iteratively select a batch of samples X_S with indexes S satisfying $S \subset U$ and $|S| = b$ until there is no budget (user specified) available for labeling, where the batch size b is a predefined constant.

The Proposed Algorithm

For a candidate batch $S \subset U$, we define $R(S)$ and $C(S)$ to be the representativeness and certainty function of set S.

We ignore the certainty score of some low-certainty samples by ensuring a lower bound on the certainty score of all unlabeled instances. Formally, for each instance $\mathbf{x}_i \in X_U$ we propose the lower-bounded certainty (LBC) score as follows

$$LC(\mathbf{x}_i) = \max(C(\mathbf{x}_i), \epsilon) \tag{1}$$

with threshold ϵ indicating the smallest accurate certainty. We mainly consider the most widely-used type of certainty, i.e., additive certainty, with the following definition.

Definition 1. ***Additive Certainty:*** *We say certainty function $C(\cdot) : 2^U \mapsto \mathcal{R}$ to be additive iff $C(S) = \sum_{i \in S} C(\mathbf{x}_i)$ for any $S \subset U$.*

Additivity indicates that the certainty measure is calculated in the instance level, and it applies to most of the certainty measures, such as entropy (Chakraborty et al. 2015b) and margin sampling (Scheffer, Decomain, and Wrobel 2001).

With this nice property, we define the LBC on a set $S \subset U$ as

$$LC(S) = \sum_{i \in S} \max(C(\mathbf{x}_i), \epsilon)$$

Without loss of generality, we directly combine the representativeness $R(S)$ with LBC score $LC(S)$ and obtain the BMAL framework with LBC:

$$\min_{S \subset U, |S|=b} R(S) + \lambda LC(S) \tag{2}$$

where λ is a trade-off parameter.

Before instantiating our method with specific representativeness and certainty, we prove that the direct and screening approaches are two special cases of our method in Eq. (2).

Lemma 1. *For representativeness $R(S)$ and additive certainty $C(S)$, the direct approach with the following objective*

$$\min_{S \subset U, |S|=b} R(S) + \lambda C(S), \tag{3}$$

and the screening approach that optimizes

$$\begin{aligned} &\min_{S \subset U, |S|=b} R(S) \\ &s.t.\ C(\mathbf{x}_i) \leq \epsilon \textit{ for all } i \in S \end{aligned} \tag{4}$$

are both special cases of our method in Eq. (2)*.*

Proof. **the direct approach:** the direct approach directly combines representativeness with uncertainty. By setting threshold ϵ to $\min_{i \in U} C(\mathbf{x}_i)$, we have $C(\mathbf{x}_i) \geq \epsilon$ for all $i \in U$, i.e. $\max(C(\mathbf{x}_i), \epsilon) = C(\mathbf{x}_i)$. The LBC $LC(\cdot)$ degenerates to certainty $C(\cdot)$, and thus Eq. (2) becomes equivalent to Eq. (3).

the screening approach: The screening approach minimizes merely representativeness over uncertain samples. Let S^* be the global optimizer of the screening approach in Eq. (4), and let v^* be the smallest violation in certainty, i.e., $v^* = \min_{C(\mathbf{x}_i) > \epsilon} C(\mathbf{x}_i) - \epsilon$ for $i \in U$. When the trade-off parameter λ satisfies $\lambda > R(S^*)/v^*$, any instance $\mathbf{x}_j$ with $c(\mathbf{x}_j) > \epsilon$ will be heavily penalized such that for any S satisfying $j \in S$, we have $f(S) = R(S) + \lambda LC(S) \geq R(S) + \lambda LC(\mathbf{x}_j) \geq R(S) + (R(S^*)/v^*)C(\mathbf{x}_j) > R(S) + R(S^*) \geq R(S^*) = R(S^*) + \lambda LC(S^*) = f(S^*)$, i.e. any batch of samples that contains the instance whose certainty is above ϵ is not the global optimal of Eq. (2). Therefore, the solution to Eq. (2) is restricted to the instances with certainty lower than or equal to ϵ, i.e. optimizing the objective in Eq. (2) is equivalent to optimizing Eq. (4) when λ becomes large. □

The above lemma holds for most of the recent BMAL algorithms that utilize both representativeness and (un)certainty. (Wang and Ye 2015; Chakraborty et al. 2015a; 2015b; Chattopadhyay et al. 2012; Yang et al. 2015).

In the following, we instantiate our algorithm using specific certainty and representativeness. For a probabilistic classifier with parameter $\mathbf{w}$, we assume the true labels of unlabeled samples are drawn from categorical distributions, i.e

$$y_i \sim Categorical(\mathbf{p}_i) \tag{5}$$

where $\mathbf{p}_i$ is the vector of the probability estimate of $\mathbf{x}_i$

$$\mathbf{p}_i = [P(y = 0|\mathbf{x}_i, \mathbf{w}); ...; P(y = |y||\mathbf{x}_i, \mathbf{w})] \tag{6}$$

In this paper, we use accuracy as the evaluation metric. Therefore, the expected accuracy on the unlabeled set U becomes

$$\widehat{ACC}_U = E_{y_i \sim Cat(\mathbf{p}_i), i \in U} \frac{1}{|U|} I(y_i = \hat{y}_i) \tag{7}$$

where $\hat{y}_i = \arg\max_y P(y|\mathbf{x}_i, \mathbf{w})$ is the predicted label. We can easily calculate the expected accuracy as

$$\widehat{ACC}_U = \frac{1}{|U|} \max_y P(y|\mathbf{x}_i, \mathbf{w}) \tag{8}$$

Following (Wang et al. 2018), the uncertainty of $\mathbf{x}_i$ is defined as the expected accuracy on $U \backslash i$. After droping constants, the uncertainty of sample $\mathbf{x}_i$ becomes

$$C(\mathbf{x}_i) = \widehat{ACC}_{U \backslash i} \sim \max_y P(y|\mathbf{x}_i, \mathbf{w}) \tag{9}$$

In the initial stage of BMAL, the certainty measure is usually noisy because there is no adequate labeled data to train the classifier. Nevertheless, samples with high certainty usually remain certain under a noisy classifier (trained with inadequate data). To be more specific, the instances with high certainty are generally far away from the decision boundary, so they are highly likely to remain certain whether under a noisy classifier or the ground-truth classifier (trained with all data). As a result, even a noisy classifier predicts accurately the highly certain samples. On the other hand, uncertain samples are usually close to the decision boundary, which makes their certainty scores untrustworthy under a noisy classifier with an inaccurate decision boundary. To alleviate this issue, we bound the certainty scores from below

$$LC(S) = \sum_{i \in S} \max(C(\mathbf{x}_i), \epsilon) \tag{10}$$

In the beginning of the sample selection process, so-called representativeness are critical because it captures the global structure of the unlabeled data. We use a typical representativeness measure named empirical MMD score (See (Gretton et al. 2006)(Gretton et al. 2012) for more details), which selects instances that minimizes the difference in distribution between labeled and unlabeled data:

$$R(S) := \mathbf{MMD}[\phi, X_{L\cup S}, X_{U\setminus S}] = \frac{1}{2}\sum_{i\in S}\sum_{j\in S} K_{ij} + \sum_{i\in S} h_i$$
$$h_i = \frac{n_u - b}{n}\sum_{j\in L} K_{ij} - \frac{n_l + b}{n}\sum_{j\in U} K_{ij} \tag{11}$$

where $\phi(\cdot)$ is a kernel mapping and K is the kernel matrix that measures similarity over all instances.

After substituting Eq. (11) and Eq. (10) into Eq. (2), we have

$$\min_{S\subset U, |S|=b} f(S) = \frac{1}{2}\sum_{i\in S}\sum_{j\in S} K_{ij} + \sum_{i\in S} h_i + \lambda\sum_{i\in S} LC(\mathbf{x}_i) \tag{12}$$

We call Eq. (12) an example of *BMAL with lower bounded certainty (LBC)*. For this instance of BMAL with LBC, we further prove that under mild assumptions, the objective function in Eq. (12) is a super-modular function. For simplicity, we define $f_T(e) := f(T\cup\{e\}) - f(T)$, for any $T \subset U$ and $e \in U\setminus T$.

Lemma 2. *For a non-negative kernel Gram matrix K such that $K_{ij} \geq 0$, the objective function $f(\cdot)$ in Eq.* (12) *is a super-modular set function defined on 2^U.*

Proof. By the definition of super-modularity functions, for any $S \subset T \subset U$ and any $e \in U\setminus T$, we have $f_T(e) - f_S(e) = \sum_{j\in T\setminus S} K_{ej} \geq 0$, thus completing the proof. □

Solving the Objective

Trivial Solution. Previous work (Chattopadhyay et al. 2012) solves an optimization problem by minimizing MMD in Eq. (11) using Quadratic Programming (QP). We point out a draw-back of the QP solver that it may give a trivial solution when the initial labeled set L is empty. The optimization problem is obtained by giving each instance $\mathbf{x}_i$ an indicator variable $\alpha_i \in \{0, 1\}$ and relaxing it to $[0, 1]$ as follows.

$$\min_{0\leq\alpha_i\leq 1, \sum_i \alpha_i = b} \frac{1}{2}\boldsymbol{\alpha}^T K_{UU}\boldsymbol{\alpha} - \frac{b}{n}\sum_{i\in U}\sum_{j\in U} K_{ij}\alpha_i \tag{13}$$

where K_{UU} is a sub-matrix of K over U. The KKT conditions of the above convex optimization problem becomes

$$\begin{cases} 0 \leq \alpha_i \leq 1 \\ \sum_i \alpha_i = b \\ \mu_i(\alpha_i - 1) = 0 \\ \boldsymbol{\mu} \geq 0, \boldsymbol{\omega} \geq 0 \\ \omega_i\alpha_i = 0 \\ \sum_{j\in U}(\alpha_j - \frac{b}{n})K_{ji} + \sum_i \mu_i + \sum_i \omega_i + t = 0 \end{cases} \tag{14}$$

where t, $\boldsymbol{\mu} \geq 0$ and $\boldsymbol{\omega} \geq 0$ are dual variables.

We can easily verify that $\alpha_i = b/n$ ($t = 0, \boldsymbol{\mu} = 0, \boldsymbol{\omega} = 0$) is a solution of the above KKT conditions, and thus a global optimal of Eq. (13). This solution gives equal weights to each unlabeled instance, resulting in undesired behavior similar to random sampling. To avoid this, we use a random greedy solver to solve our similar objective in Eq. (12)

Random Greedy Solver. The random greedy algorithm (Buchbinder et al. 2014) maximizes a nonnegative sub-modular function with cardinality constraint. Since minimizing super-modular functions is equivalent to maximizing sub-modular functions, this algorithm can also be applied to our objective function in Eq. (12). It starts with an empty set, and in each iteration a set of possible "good" indexes is constructed, which consists of b instances that increase the objective function least. One index is then randomly selected from the set of "good" indexes and is added to the solution. This process is repeated until b instances are selected.

The increase of the objective function $f(\cdot)$ after adding index e to set S can be formulated as

$$f_S(e) = f(S\cup\{e\}) - f(S) = \sum_{i\in S} K_{ie} + \frac{1}{2}K_{ee} + h_e + \lambda LC(\mathbf{x}_e)$$

After selecting another index e', the increase becomes

$$f_{S\setminus e'}(e) = \sum_{i\in S\setminus e'} K_{ie} + \frac{1}{2}K_{ee} + h_e + \lambda LC(\mathbf{x}_e) = f_S(e) - K_{ee'}$$

Algorithm 1 summarizes the random greedy algorithm, where ψ and ψ' correspond to f_S and $f_{S\setminus e'}$ respectively.

Algorithm 1 RandGreedy(U,b)% select b instances from U

Require: $\mathbf{h}$, kernel K, batch size b, unlabeled index set U
Ensure: A solution S
1: $S \leftarrow \emptyset$
2: $\psi_e = \frac{1}{2}K_{ee} + h_e + \lambda LC(\mathbf{x}_e)$, for all $e \in U$
3: **for** i=1 to b **do**
4: Let M be the set of b indexes in $U\setminus S$ with b smallest ψ value
5: Randomly select one index e' from M
6: $S \leftarrow S\cup\{e'\}$
7: $\psi'_e \leftarrow \psi_e - K_{ee'}$, for all $e \in U\setminus S$
8: $\psi_e \leftarrow \psi'_e$, for all $e \in U\setminus S$
9: **end for**
10: **return** S

Coefficient Between Batches. When one batch of samples is selected for labeling, we have to update the coefficient $\mathbf{h}$ in Eq. (11) according to the selected instances. Let L^t be the labeled index set L at the t-th batch ($t = 1, 2, ...$). Similar notations apply for n_u, n_l, U and $\mathbf{h}$. We split Eq. (11) as follows.

$$h_i^t = \frac{n_u^t - b}{n}\sum_{j\in L^t} K_{ij} - \frac{n_l^t + b}{n}\sum_{j\in U^t} K_{ij} \tag{15}$$

For n_u and n_l, we have $n_u^{t+1} = n_u^t - b$ and $n_l^{t+1} = n_l^t + b$.

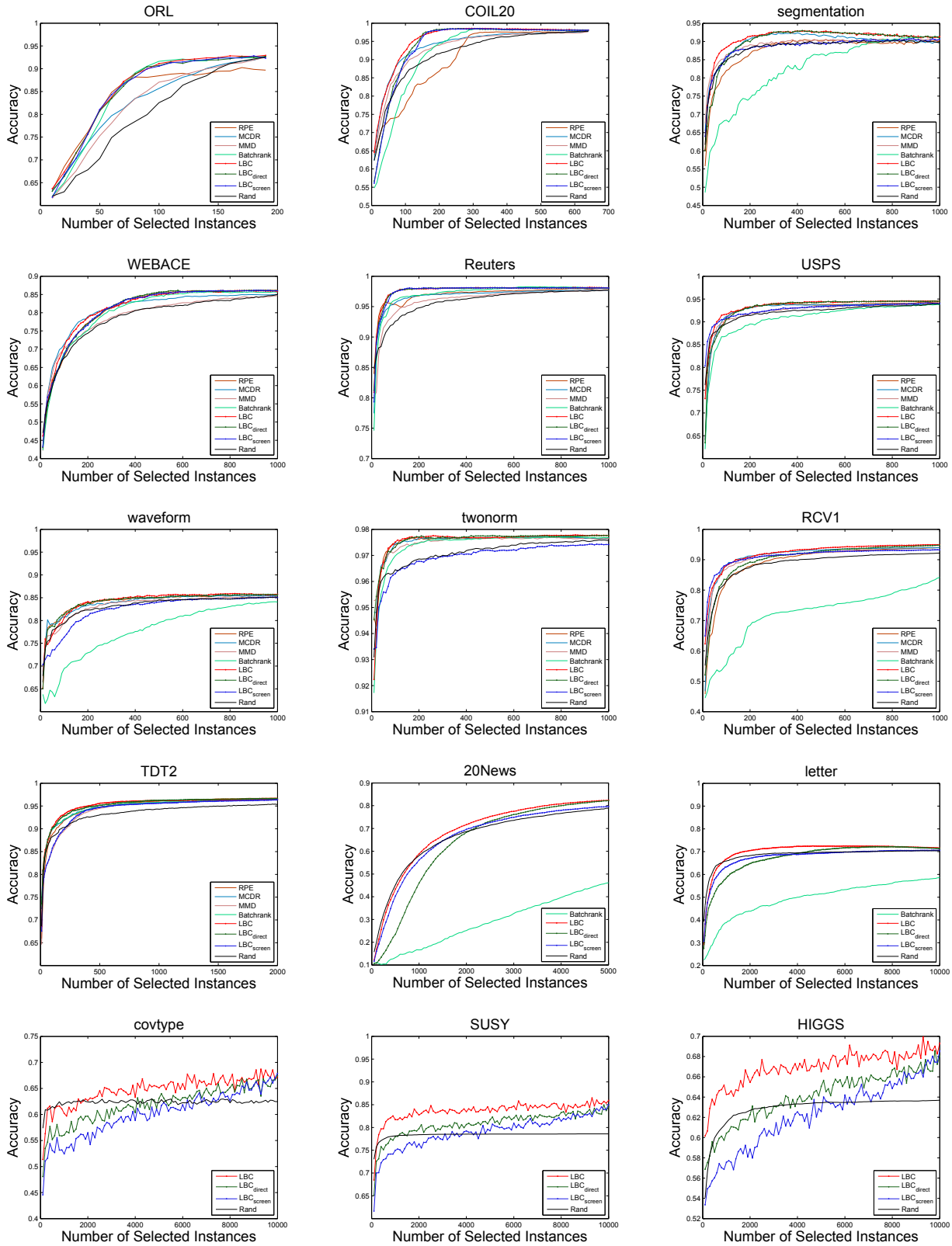

Figure 1: Accuracy of LBC against seven baselines over fifteen datasets .

We have the following recursive formula to calculate h^{t+1}

$$h_i^{t+1} = h_i^t - \frac{b}{n}\sum_{j\in D} K_{ij} + \frac{nu^t - n_l^t - 4b}{n}\sum_{j\in S^t} K_{ij} \quad (16)$$

Calculating coefficient $\mathbf{h}$ using Eq. (15) and Eq. (16), we obtain Algorithm 2 for BMAL based on LBC.

Algorithm 2 BMAL based on LBC

Require: batch size b, labeled index set L, unlabeled index set U, and data matrix X, batch number t
Ensure: A solution S of size b
1: **if** t=1 **then**
2: Construct Kernel matrix K from X
3: Initialize coefficient $\mathbf{h}^1$ using Eq. (15)
4: **end if**
5: $S \leftarrow \text{RandGreedy}(U, b)$
6: Update coefficient $\mathbf{h}^{t+1}$ with $\mathbf{h}^t$ using Eq. (15) and Eq. (16)
7: **return** S

Memory-Efficient BMAL. Note that the random greedy method (Algorithm 1) requires the input to be an $n \times n$ kernel matrix, which is difficult to store when the number of instances n becomes large. In a simple case where $d \ll n$ and the $n \times d$ data matrix can fit into memory, we only need to calculate the similarity matrix between (un)labeled set L (U) and selected instances S.

To reduce time complexity, we adopt the technique named Random Fourier Features (RFF) (Chitta, Jin, and Jain 2012) to calculate h^t with a low-rank representation of the data. The kernel function $K(\mathbf{x}, \mathbf{y})$ can be approximated as $K(\mathbf{x}, \mathbf{y}) = \mathbf{R}(\mathbf{x})\mathbf{R}(\mathbf{y})^T$ where $\mathbf{R}(\mathbf{x})$ is the low-rank representation of $\mathbf{x}$.

Using Eq. (15) and Eq. (16) as well as the RFF technique to calculate coefficient $\mathbf{h}$, and directly calling the kernel function when needed, we obtain the memory-efficient version of Algorithm 2. The time complexity of Algorithm 2 is $\mathcal{O}(n_u b)$. The memory-efficient version runs in $\mathcal{O}(n_u b D)$, where D is the number of Fourier components in RFF. Algorithm 2 requires $\mathcal{O}(n_u^2)$ space, while the memory efficient version only needs $\mathcal{O}(nD)$.

Complexity Analysis. The random greedy algorithm in Algorithm 1 runs in $\mathcal{O}(n_u b)$. Updating coefficient h requires $\mathcal{O}(n^2)$ time in the first batch, and $\mathcal{O}(n_u b)$ otherwise. Finally, constructing the kernel matrix usually takes $\mathcal{O}(n^2 d)$ time in the first batch. To sum up, the time complexity[1] of Algorithm 2 is $\mathcal{O}(n^2 d)$ when the batch number $t = 1$ and $\mathcal{O}(n_u b)$ otherwise.

The memory-efficient version of Algorithm 2 does not store or pre-calculate the kernel matrix, and the kernel function is activated only when called upon. In the first batch, calculating coefficient $\mathbf{h}$ requires $\mathcal{O}(nDd)$ using RFF. Therefore, the total running time of the memory-efficient method becomes $\mathcal{O}(ndD + nbd)$ when $t = 1$ and $\mathcal{O}(n_u b D)$ otherwise. The additional memory is reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(nD)$, where D is the number of Fourier components. We empirically set $D = 100$ in our experiment.

[1]Here we do not consider the time complexity to obtain the certainty score, which is specified by the classifier

Name	Number of Instances	Number of Features	Number of Classes
ORL	400	1024	40
COIL20	1440	1024	20
segmentation	2310	19	7
WEBACE	2340	1000	20
Reuters	2919	18933	4
USPS	3082	256	4
waveform	5000	21	3
twonorm	7400	20	2
RCV1	9625	29992	4
TDT2	10212	36771	96
20News	18774	61188	20
letter	20000	16	26
covtype	581012	54	7
SUSY	5×10^6	18	2
HIGGS	1.1×10^7	28	2

Table 1: Dataset Description

Experimental Results

Datasets We use fifteen benchmark datasets, seven of which are from UCI machine learning repository (Dheeru and Karra Taniskidou 2017), namely *segmentation*, *waveform*, *twonorm*, *HIGGS*, *covtype*, *SUSY* and *letter*. The other eight datasets are *Reuters*, *RCV1*, *TDT2*, *20News*, *WEBACE*, *ORL*, *COIL20* and *USPS*, which are publicly available[2]. The tasks of these datasets range from hand writing digits recognition, object recognition, to text classification. Table 1 summarizes the details of the datasets.

Experiment Setup

All the compared methods are described below.

- **LBC**: our BMAL method in Algorithm 2.
- **LBC**$_{direct}$: which is a degenerated version of our method by taking the direct approach.
- **LBC**$_{screen}$: which is another degenerated version of our method by taking the screening approach.
- **Batchrank**: the BMAL method proposed in (Chakraborty et al. 2015b) which directly combines mutual information with entropy.
- **RPE**: the representative BMAL method described in (Wang et al. 2015) using relative Pearson divergence.
- **MMD**: the QP-based MMD method (Chattopadhyay et al. 2012), which selects samples using QP relaxation.
- **MCDR**: the BMAL method in (Wang and Ye 2015) which directly combines MMD with a regression loss
- **Rand**: random selection, which selects b samples uniformly at random.

In the experiment, we randomly split each dataset into unlabeled data (60%) and testing data (40%). One instance from each class is randomly selected as the initial labeled

[2]http://www.cad.zju.edu.cn/home/dengcai/

	Time(s)					acc.ratio			
Dataset	MMD	RPE	Batchrank	MCDR	LBC	vs. MMD	vs. RPE	vs. Batchrank	vs. MCDR
ORL	0.35	0.21	0.28	2.81	0.01	63.88	38.38	51.41	513.30
COIL20	37.84	19.37	8.70	59.52	0.16	234.97	120.28	54.04	369.58
segmentation	240.34	82.02	13.06	165.93	0.46	523.09	178.52	28.43	361.14
WEBACE	188.12	77.48	25.81	158.67	0.49	387.41	159.56	53.14	326.77
Reuters	539.71	99.73	16.76	384.22	0.83	647.73	119.69	20.12	461.12
USPS	368.49	133.81	19.83	147.28	0.89	414.07	150.37	22.29	165.50
waveform	3131.24	345.51	42.55	204.78	2.51	1245.58	137.44	16.93	81.46
twonorm	78.35	737.10	73.81	191.39	5.68	13.80	129.85	13.00	33.72
RCV1	7458.11	1061.75	184.23	11320.09	9.66	772.08	109.91	19.07	1171.88
TDT2	71093.94	239270.75	3567.2	5251.77	195.31	364.01	1225.05	18.26	26.89
20News	NA	NA	12951.53	NA	333.45			33.84	
letter	NA	NA	25325.87	NA	598.61			42.31	
covtype	MLE	MLE	MLE	MLE	10296.35				
SUSY	MLE	MLE	MLE	MLE	59595.7				
HIGGS	MLE	MLE	MLE	MLE	159035.40				

Table 2: Total running time (in seconds) of five compared methods along with accelerating ratio of *LBC* against the other four. NA represents the method fails to provide with a result after running for several days, and MLE indicates that the method runs out of memory.

	Win/Loss(%) of LBC vs. the following						
Dataset	Rand	MMD	RPE	Batchrank	MCDR	direct	screen
ORL	74/0	53/0	53/0	0/0	53/0	0/0	0/0
COIL20	86/0	90/0	54/0	44/0	71/0	20/0	21/0
segmetation	96/0	84/2	98/0	81/0	66/0	11/0	91/0
WEBACE	91/0	89/0	11/1	20/0	52/3	12/1	7/0
Reuters	98/1	99/0	66/0	47/0	98/0	0/0	1/0
USPS	96/1	84/0	32/0	100/0	64/0	9/0	82/4
waveform	85/0	76/0	3/4	99/0	13/4	0/4	90/0
twonorm	78/0	10/2	0/0	11/0	1/0	0/1	97/0
RCV1	99/0	90/0	100/0	100/0	58/0	91/0	78/5
TDT2	97/0	53/0	40/0	44/0	53/0	9/0	72/0
20News	80/14	NA	NA	100/0	NA	92/0	98/0
letter	90/6	NA	NA	100/0	NA	74/0	98/0
covtype	32/2	MLE	MLE	MLE	MLE	3/0	37/0
SUSY	98/1	MLE	MLE	MLE	MLE	89/0	98/0
HIGGS	100/0	MLE	MLE	MLE	MLE	2/0	11/0

Table 3: the win/loss(%) of *LBC* against BMAL baselines using paired t-test with a 95% significant level

data. All methods are applied with the same initial, unlabeled and testing dataset. The batch size b is fixed to be 100 on large datasets *covtype*, *SUSY* and *HIGGS*, 50 on *letter* and *20News*, and 10 on other small datasets. Logistic Regression is used as the classifier. For each dataset, the experiment is conducted 10 times. The averaged result is reported.

We use Gaussian kernel on all datasets. For data instances $\mathbf{x}$ and $\mathbf{y}$ we set $K(\mathbf{x}, \mathbf{y}) = exp(-||\mathbf{x} - \mathbf{y}||^2/p)$, where the parameter p is the *median* of all pair-wise squared Euclidean distances over the unlabeled data. We sort all the unlabeled samples increasingly according to their certainty in Eq. (10), and set hyper-parameter ϵ to be the β-th percentile ($0 < \beta < 100$). We empirically use two hyper-parameter γ and τ to describe β as $\beta = \gamma * (n_u/n)^\tau$, where γ and τ is fixed to be 20 and 10 respectively. For hyper-parameter λ, we set $\lambda = b^2$. The two degenerated methods use the same hyper-parameters as our method. For the three largest datasets, we use the memory-efficient version of Algorithm 2 instead.

Experimental Results

Running Time. Table 2 shows the total running time of five compared methods, along with the accelerating ratio of *LBC* against the other four. For datasets such as *20News* and *letter*, some baselines are omitted because they cannot present the results after running several days. For datasets *covtype*, *HIGGS* and *SUSY*, where the number of unlabeled instances reaches 10^5 to 10^6, four baselines run out of memory. As can be seen, *LBC* has smaller running time on all datasets against the four complex baselines. It is interesting to investigate the accelerating ratio of *LBC* against *MMD* since they are solving similar objective using different solvers. For datasets *RCV1*, *LBC* is over 700 times faster than *MMD*, and for dataset *waveform*, the accelerating ratio is over 1200.

Accuracy. Figure 1 shows the average accuracy of all compared methods over fifteen datasets. We can see that our algorithm at least does not lose accuracy from the figures. Table 3 further reveals the percentage of win/loss of *LBC* against five baselines using paired t-test with $p < 0.05$. The t-tests are conducted on the accuracy of compared methods over 10 runs. As we can see, *LBC* wins most of the batches on most datasets. Our method ties against *RPE* on dataset *twonorm*, and also ties against *batchrank* on *ORL*. It also becomes slightly worse in accuracy on dataset *waveform* and *twonorm* against LBC_{direct}. In *Reuters* and *ORL*, the two degenerated versions have similar performance with our method.

Conclusion

In this paper we propose a generalized algorithm that demonstrates the connection between the direct and screening method. We use MMD and LBC as a special case of representativeness and certainty for better empirical results. The objective is efficiently solved using a random greedy algorithm that avoids the trivial solution induced by the original QP solver. Experiments on fifteen datasets demonstrate that while having significantly higher accuracy, our method also scales better than the latest state-of-the-art BMAL methods.

Acknowledgments

This work is supported in part by China National 973 program 2014CB340301.

References

Buchbinder, N.; Feldman, M.; Naor, J. S.; and Schwartz, R. 2014. Submodular maximization with cardinality constraints. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1433–1452. SIAM.

Chakraborty, S.; Balasubramanian, V.; Sankar, A. R.; Panchanathan, S.; and Ye, J. 2015a. Batchrank: A novel batch mode active learning framework for hierarchical classification. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

Chakraborty, S.; Nallure Balasubramanian, V.; Sun, Q.; Panchanathan, S.; and Ye, J. 2015b. Active batch selection via convex relaxations with guaranteed solution bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(10):1945 – 1958.

Chattopadhyay, R.; Wang, Z.; Fan, W.; Davidson, I.; Panchanathan, S.; and Ye, J. 2012. Batch mode active sampling based on marginal probability distribution matching. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 741–749.

Chen, Y., and Krause, A. 2013. Near-optimal batch mode active learning and adaptive submodular optimization. In *Proceedings of the 30th International Conference on Machine Learning*, 160–168.

Cheng, Y.; Chen, Z.; Fei, H.; Wang, F.; and Choudhary, A. N. 2014. Batch mode active learning with hierarchical-structured embedded variance. In *SIAM International Conference on Data Mining*, 10–18. SIAM.

Chitta, R.; Jin, R.; and Jain, A. K. 2012. Efficient kernel clustering using random fourier features. In *2012 IEEE 12th International Conference on Data Mining*, 161–170. IEEE.

Dheeru, D., and Karra Taniskidou, E. 2017. UCI machine learning repository.

Gretton, A.; Borgwardt, K. M.; Rasch, M.; Schölkopf, B.; and Smola, A. J. 2006. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems*, 513–520.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A kernel two-sample test. *The Journal of Machine Learning Research* 13(1):723–773.

Gu, Q.; Zhang, T.; and Han, J. 2014. Batch-mode active learning via error bound minimization. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, 300–309.

Guo, Y., and Schuurmans, D. 2008. Discriminative batch mode active learning. In *Advances in Neural Information Processing Systems*, 593–600.

Guo, Y. 2010. Active instance sampling via matrix partition. In *Advances in Neural Information Processing Systems*, 802–810.

Hoi, S. C.; Jin, R.; Zhu, J.; and Lyu, M. R. 2006. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd International Conference on Machine Learning*, 417–424.

Reyes, O., and Ventura, S. 2018. Evolutionary strategy to perform batch-mode active learning on multi-label data. *ACM Transactions on Intelligent Systems and Technology (TIST)* 9(4):46.

Scheffer, T.; Decomain, C.; and Wrobel, S. 2001. Active hidden markov models for information extraction. In *International Symposium on Intelligent Data Analysis*, 309–318. Springer.

Settles, B. 2010. Active learning literature survey. *University of Wisconsin, Madison* 52:55–66.

Seung, H. S.; Opper, M.; and Sompolinsky, H. 1992. Query by committee. In *Proceedings of the 5th annual workshop on Computational learning theory*, 287–294. ACM.

Tong, S., and Koller, D. 2002. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research* 2:45–66.

Wang, Z., and Ye, J. 2015. Querying discriminative and representative samples for batch mode active learning. *ACM Transactions on Knowledge Discovery from Data* 9(3):17:1–17:23.

Wang, H.; Du, L.; Zhou, P.; Shi, L.; and Shen, Y.-D. 2015. Convex batch mode active sampling via α-relative pearson divergence. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*.

Wang, H.; Chang, X.; Shi, L.; Yang, Y.; and Shen, Y.-D. 2018. Uncertainty sampling for action recognition via maximizing expected average precision. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 964–970.

Wei, K.; Iyer, R.; and Bilmes, J. 2015. Submodularity in data subset selection and active learning. In *Proceedings of the 32nd International Conference on Machine Learning*, 1954–1963.

Yan, Y., and Huang, S.-J. 2018. Cost-effective active learning for hierarchical multi-label classification. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2962–2968.

Yang, Y.; Ma, Z.; Nie, F.; Chang, X.; and Hauptmann, A. G. 2015. Multi-class active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision* 113(2):113–127.