Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Convex Batch Mode Active Sampling via α-relative Pearson Divergence

Hanmo Wang[1,2] **, Liang Du**[1] **, Peng Zhou**[1,2] **, Lei Shi**[1,2] and **Yi-Dong Shen**[1*]
[1]State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
{wanghm,duliang,zhoup,shilei,ydshen}@ios.ac.cn

Abstract

Active learning is a machine learning technique that trains a classifier after selecting a subset from an unlabeled dataset for labeling and using the selected data for training. Recently, batch mode active learning, which selects a batch of samples to label in parallel, has attracted a lot of attention. Its challenge lies in the choice of criteria used for guiding the search of the optimal batch. In this paper, we propose a novel approach to selecting the optimal batch of queries by minimizing the α-relative Pearson divergence (RPE) between the labeled and the original datasets. This particular divergence is chosen since it can distinguish the optimal batch more easily than other measures especially when available candidates are similar. The proposed objective is a min-max optimization problem, and it is difficult to solve due to the involvement of both minimization and maximization. We find that the objective has an equivalent convex form, and thus a global optimal solution can be obtained. Then the subgradient method can be applied to solve the simplified convex problem. Our empirical studies on UCI datasets demonstrate the effectiveness of the proposed approach compared with the state-of-the-art batch mode active learning methods.

Introduction

Active learning is proposed to alleviate the effort of the labeling process by selecting informative data samples. It is useful when unlabeled data are abundant but manual labeling is expensive. The challenge of active learning is that, given a large pool of unlabeled data and a relatively small labeling budget, the classifier trained on selected labeled data must have good generalization performance on unseen data. In other words, an active learning algorithm selects only a few data instances for labeling while maintaining certain classification performance.

Traditional active learning approaches that select the single most informative data example usually retrain the classifier when a new instance is labeled. Under circumstances where multiple annotators are working concurrently, batch mode active learning which iteratively selects a batch of queries to label is more efficient and appropriate. In the batch mode active learning process, the learner is given a labeled set and an unlabeled set, and iteratively chooses a batch of instances from the unlabeled set to query for the labels. The main difficulty of batch active learning is under what criterion the batch is selected.

One of the most recent work in batch mode active learning uses representative information based on distribution matching (Chattopadhyay et al. 2013). It adopts maximum mean discrepancy (MMD) (Gretton et al. 2006) to select the batch that minimizes the empirical MMD score between labeled and unlabeled data. It turns out that the use of MMD to capture representative information cannot effectively distinguish between the optimal batch and the other candidates (Settles 2010; Wang and Ye 2013). When more data are labeled, the induced candidate batches become more similar, which makes the problem of MMD seems more serious. Therefore, to handle this issue, we propose to use an alternative measure called α-relative Pearson divergence (RPE) (Yamada et al. 2011) which is more appropriate to compare distributions because of the following distinct properties. First, divergence, such as K-L divergence (Kullback and Leibler 1951), is well-known to be suitable for distribution comparison. Second, the superiority of RPE against MMD has been shown in two-sample distribution matching. Therefore, the effectiveness of RPE in distribution-oriented batch active learning could also be expected. In addition, it is also shown that RPE often gives larger dissimilarity score when two distributions are similar but non-identical (Yamada et al. 2011). It is with the two nice properties that RPE is usually able to distinguish between the optimal batch and the other candidates. Such advantages are further enlarged when the distributions represented by different candidates become more similar.

Our main contributions can be summarized as follows.

1. We propose a novel batch mode active learning algorithm based on α-relative Pearson divergence (RPE) whose properties are suitable for this task.

2. When using RPE in batch active learning, some auxiliary variable will be introduced. As a result, the overall objective function becomes a min-max optimization problem, and generally it is hard to solve because the objective function is simultaneously maximized and minimized with respect to the first and second variable. To address

*corresponding author

this issue, we prove that the problem has an equivalent convex formulation after swapping the two variables, so it can be solved by applying existing convex programming methods.

3. Our empirical studies on 6 UCI datasets show that the proposed method, namely RPE_{active}, significantly outperforms the state-of-the-art batch mode active learning algorithms in general.

Related Work

Active Learning

Different methods for active learning are proposed in the last decades (Settles 2010). Some of the popular approaches select the single most informative data point in each iteration. One of the single-instance approaches is the uncertain sampling method, which chooses the most uncertain instance to label at each iteration. Uncertainty can be measured by the distance to decision boundary (Campbell et al. 2000; Schohn and Cohn 2000; Tong and Koller 2002) or entropy of predicted label (Settles and Craven 2008). Another popular single-instance active learning method is query-by-committee (Seung, Opper, and Sompolinsky 1992; Dagan and Engelson 1995; Freund et al. 1997);it trains multiple classifiers and selects the data instance on which the classifiers have the most disagreement.

Batch Mode Active Learning

In recent years, various criteria are proposed to select the most informative batch of samples. (Guo 2010) proposes an approach to select the batch that minimizes the mutual information between labeled and unlabeled data, (Yu, Bi, and Tresp 2006) chooses data points that have the lowest linear-reconstruction error, (Hoi et al. 2006) applies the Fisher information matrix to select the optimal batch, and (Guo and Schuurmans 2008) proposes a discriminate approach. Most recently, (Chattopadhyay et al. 2013) proposes a method to minimize the difference in distribution between labeled and unlabeled data, after selecting a batch. (Wang and Ye 2013) further combines the distribution-matching method with discriminative information.

Preliminaries

In this section, we briefly introduce the batch mode active learning setting and the α-relative Pearson divergence.

Problem Setting Suppose we have a d-dimensional dataset $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$ of n data points. Let the set $U = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_{n_u}\}$ be the set of unlabeled data and let $L = \{\boldsymbol{x}_{n_u+1}, ..., \boldsymbol{x}_{n_u+n_l}\}$ ($n_u + n_l = n$) be the set of labeled data, where n_l and n_u denote the number of labeled data and unlabeled data respectively. Additionally, labeled dataset L is associated with labels $Y_l = \{y_{n_u+1}, ..., y_n\} \in \{-1, 1\}^{n_l}$. For a predefined batch size n_s, the goal of batch mode active learning is to select the best batch S with $S \subseteq U$ and $|S| = n_s$ for labeling such that the classifier trained from labeled data has low generalization error on unseen data i.i.d. drawn from the same distribution as D. The algorithm should choose S iteratively until a given budget of labels is reached.

In this paper, we use $\mathcal{K}(\cdot, \cdot)$ to denote the kernel function and $\boldsymbol{K}_{ij} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ to denote the kernel Gram matrix. We also denote $\boldsymbol{K}_{XL} = \boldsymbol{K}(1 : n, n_u + 1 : n)$ with MATLAB notations and $\boldsymbol{K}_{LX} = (\boldsymbol{K}_{XL})^T$ where $(\cdot)^T$ is the matrix transpose operator. Similarly $\boldsymbol{K}_{XU} = \boldsymbol{K}(1 : n, 1 : n_u)$, and $\boldsymbol{K}_{UX} = (\boldsymbol{K}_{XU})^T$. $\mathbf{1}_n$ is a length n column vector of all 1s. $diag(\boldsymbol{\beta})$ is denoted as the diagonal matrix where $\boldsymbol{\beta}$ is its main diagonal. We also denote $\boldsymbol{K}_{i*}$ and and $\boldsymbol{K}_{*i}$ as the i-th row and the i-th column of matrix $\boldsymbol{K}$.

α-relative Pearson divergence We briefly introduce the α-relative Pearson divergence for two sample test. For two unknown d-dimensional distribution P and Q, we have $V = \{\boldsymbol{v}_i\}_{i=1}^{n_v}$ i.i.d. drawn from P and $T = \{\boldsymbol{t}_j\}_{j=1}^{n_t}$ i.i.d. drawn from Q. The α-relative Pearson divergence is an estimate of the similarity between P and Q given only V and T. Let $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ be the probability density functions of P and Q, respectively.

Intuitively, after calculating the density ratio $r(\boldsymbol{x}) = p(\boldsymbol{x})/q(\boldsymbol{x})$, the goal of RPE is to compute the Pearson divergence $PE(P, Q) = \mathbb{E}_{q(\boldsymbol{x})}[(r(\boldsymbol{x}) - 1)^2]$ of two distributions based on two samples. Since directly calculating $r(\boldsymbol{x})$ is hard (Yamada et al. 2011), RPE uses the α-relative density-ratio $r_\alpha(\boldsymbol{x}) = p(\boldsymbol{x})/(\alpha p(\boldsymbol{x}) + (1 - \alpha)q(\boldsymbol{x}))$ and computes the α-relative Pearson divergence $PE_\alpha(P, Q) = \frac{1}{2}\mathbb{E}_{q_\alpha(\boldsymbol{x})}[(r_\alpha(\boldsymbol{x}) - 1)^2]$. At last, the estimates $\hat{r}_\alpha$ and $\widehat{PE}_\alpha$ are obtained by solving a quadratic objective and replacing expectations with averages. The details of RPE are presented as follows, and more comprehensive theoretical analysis can be found in (Yamada et al. 2011).

The estimate of $r_\alpha(\boldsymbol{x})$ is defined as a linear combination of kernel functions

$$\hat{r}_\alpha(\boldsymbol{x}) := g(\boldsymbol{x}; \hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n_v} \hat{\theta}_i \mathcal{K}(\boldsymbol{v}_i, \boldsymbol{x}) \tag{1}$$

where $\mathcal{K}(\cdot, \cdot)$ is a kernel function and $\hat{\boldsymbol{\theta}}$ is obtained by minimizing the expected squared loss

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{q_\alpha(\boldsymbol{x})}[(g(\boldsymbol{x}; \boldsymbol{\theta}) - r_\alpha(\boldsymbol{x}))^2] = \frac{\alpha}{2}\mathbb{E}_{p(\boldsymbol{x})}[g(\boldsymbol{x}; \boldsymbol{\theta})^2] + \frac{1 - \alpha}{2}\mathbb{E}_{q(\boldsymbol{x})}[g(\boldsymbol{x}; \boldsymbol{\theta})^2] - \mathbb{E}_{p(\boldsymbol{x})}[g(\boldsymbol{x}; \boldsymbol{\theta})] + const \tag{2}$$

After replacing the expectations with empirical averages in Eq. (2), $\hat{\boldsymbol{\theta}}$ is obtained by minimizing a quadratic form by adding $\frac{\lambda}{2}\boldsymbol{\theta}^T\boldsymbol{\theta}$ as a regularization term:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}}(\frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{h}^T\boldsymbol{\theta} + \frac{\lambda}{2}\boldsymbol{\theta}^T\boldsymbol{\theta}) \tag{3}$$

where

$$A_{ij} = \frac{\alpha}{n_v}\sum_{k=1}^{n_v}\mathcal{K}(\boldsymbol{v}_i, \boldsymbol{v}_k)\mathcal{K}(\boldsymbol{v}_j, \boldsymbol{v}_k) + \frac{1 - \alpha}{n_t}\sum_{k=1}^{n_t}\mathcal{K}(\boldsymbol{v}_i, \boldsymbol{t}_k)\mathcal{K}(\boldsymbol{v}_j, \boldsymbol{t}_k) \tag{4}$$

$$h_i = \frac{1}{n_v}\sum_{k=1}^{n_v}\mathcal{K}(\boldsymbol{v}_k, \boldsymbol{v}_i)$$

The α-relative Pearson divergence PE_α is defined based on the α-relative density-ratio $r_\alpha(\boldsymbol{x})$:

$$PE_\alpha(P,Q) = \frac{1}{2}\mathbb{E}_{q_\alpha(\boldsymbol{x})}[(r_\alpha(\boldsymbol{x})-1)^2] \tag{5}$$

Expanding Eq. (5) and again replacing expectations with averages, the final estimate of α-relative PE divergence can be obtained by the following definition.

Definition 1. *(Yamada et al. 2011) The estimate of α-relative PE divergence given $V = \{\boldsymbol{v}_i\}_{i=1}^{n_v}$ and $T = \{\boldsymbol{t}_j\}_{j=1}^{n_t}$ is defined as*

$$\widehat{PE}_\alpha(V,T) = -\frac{\alpha}{2n_v}\sum_{i=1}^{n_v}\hat{r}_\alpha(\boldsymbol{v}_i)^2 - \frac{1-\alpha}{2n_t}\sum_{i=1}^{n_t}\hat{r}_\alpha(\boldsymbol{t}_i)^2 + \frac{1}{n_v}\sum_{i=1}^{n_v}\hat{r}_\alpha(\boldsymbol{v}_i) - \frac{1}{2}$$

Proposed Framework

In this section, we apply the α-relative Pearson divergence to batch active learning by minimizing the divergence estimate between labeled data and the original dataset. Then we obtain a min-max objective and prove that it has an equivalent convex form. Finally, we apply the subgradient method to solve the convex objective.

Min-Max Objective

Since the α-relative Pearson divergence can choose the optimal batch among its competitors when the candidate batches are similar, we propose a batch mode active learning algorithm which iteratively selects the most representative batch of samples. More specifically, we minimize the α-relative Pearson divergence between $L \cup S$ and D at each iteration. After substituting V with D and T with $L \cup S$ in Definition 1, our objective is to solve the following problem

$$\min_{S\subseteq U, |S|=n_s} \widehat{PE}_\alpha(D, L\cup S) \tag{6}$$

We further introduce variable $\boldsymbol{\beta}$ of length n_u as the indicator variable. β_i equals 1 when $\boldsymbol{x}_i$ is selected in the batch, and 0 otherwise. Expanding Eq. (6), we reformulate it as a function $\mathcal{I}(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$.

$$\mathcal{I}(\boldsymbol{\beta}) = -\frac{\alpha}{2n}\sum_{i=1}^{n}\hat{r}_\alpha(\boldsymbol{x}_i)^2 + \frac{1}{n}\sum_{i=1}^{n}\hat{r}_\alpha(\boldsymbol{x}_i) - \frac{1-\alpha}{2(n_l+n_s)}\sum_{i=1}^{n_u}\beta_i\hat{r}_\alpha(\boldsymbol{x}_i)^2 \tag{7}$$

Then the objective (6) becomes

$$\min_{\boldsymbol{\beta}\in\{0,1\}^{n_u}, \|\boldsymbol{\beta}\|_1=n_s} \mathcal{I}(\boldsymbol{\beta}) \tag{8}$$

Note that in the third term of Eq. (7), when $\boldsymbol{x}_i$ is not selected in the batch, β_i becomes 0 and the term $\beta_i\hat{r}_\alpha(\boldsymbol{x}_i)$ is not included in the summation. Substituting Eq. (1) into Eq. (8), we get $\hat{\boldsymbol{\theta}}$ by minimizing the squared loss in Eq. (2) and replace expectations with empirical averages. After a few lines of calculation, we get

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{H}\boldsymbol{\theta} - \boldsymbol{b}^T\boldsymbol{\theta} + \frac{\lambda}{2}\boldsymbol{\theta}^T\boldsymbol{\theta} \tag{9}$$

where

$$\boldsymbol{H} = c_0(\boldsymbol{K}_{XL}\boldsymbol{K}_{LX} + \boldsymbol{K}_{XU}diag(\boldsymbol{\beta})\boldsymbol{K}_{UX}) + c_1\boldsymbol{K}^2 \tag{10}$$

and

$$\boldsymbol{b} = \frac{1}{n}\boldsymbol{K}\boldsymbol{1}_n \tag{11}$$

with $c_0 = (1-\alpha)/(n_l+n_s)$ and $c_1 = \alpha/n$. Thus the objective function in Eq. (9) can be reformulated as

$$\mathcal{J}(\boldsymbol{\theta},\boldsymbol{\beta}) = \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{H}\boldsymbol{\theta} - \boldsymbol{b}^T\boldsymbol{\theta} + \frac{\lambda}{2}\boldsymbol{\theta}^T\boldsymbol{\theta} \tag{12}$$

By substituting $\hat{r}_\alpha(\boldsymbol{x}) = \sum_{i=1}^{n}\hat{\theta}_i\mathcal{K}(\boldsymbol{x}_i,\boldsymbol{x})$ into Eq. (7) and rewriting function $\mathcal{I}(\boldsymbol{\beta})$ in matrix form, we get the relation between $\mathcal{J}(\hat{\boldsymbol{\theta}},\boldsymbol{\beta})$ and $\mathcal{I}(\boldsymbol{\beta})$ as follows:

$$\mathcal{I}(\boldsymbol{\beta}) + \mathcal{J}(\hat{\boldsymbol{\theta}},\boldsymbol{\beta}) + \frac{\lambda}{2}\hat{\boldsymbol{\theta}}^T\hat{\boldsymbol{\theta}} = 0 \tag{13}$$

Since the regularization parameter λ is usually small, we drop the regularization term $\frac{\lambda}{2}\boldsymbol{\theta}^T\boldsymbol{\theta}$ in Eq. (13) and formulate our objective (8) to a min-max optimization problem

$$\max_{\boldsymbol{\beta}\in\mathcal{B}}\min_{\boldsymbol{\theta}\in\mathcal{R}^n} \mathcal{J}(\boldsymbol{\theta},\boldsymbol{\beta}) \tag{14}$$

where $\boldsymbol{\beta}$ is relaxed from $\{0,1\}^{n_u}$ to $[0,1]^{n_u}$, and $\mathcal{B}$ is denoted as the domain of $\boldsymbol{\beta}$

$$\mathcal{B} := \{\boldsymbol{x} | 0 \le x_i \le 1, \|\boldsymbol{x}\|_1 = n_s, i = 1,2,...,n_u\} \tag{15}$$

Note that objective (14) is hard to solve because two variables are involved, and we need to maximize the objective w.r.t. $\boldsymbol{\beta}$ while minimizing it w.r.t. $\boldsymbol{\theta}$. In the next section, we present an important solution to address this issue.

Convex Reformulation

Our careful analysis reveals that the objective (14) has an *equivalent* form by eliminating variable $\boldsymbol{\beta}$. We further prove the equivalent function $\mathcal{G}(\boldsymbol{\theta})$ is a convex function. Such reformulation greatly simplifying the objective, without the involvement of the variable $\boldsymbol{\beta}$ in the optimization procedure, and the optimal $\hat{\boldsymbol{\beta}}$ can be computed by $\hat{\boldsymbol{\theta}}$.

We first prove that the variables $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ can be swapped in the min-max optimization problem (14).

Lemma 1. *The min and max operations in Eq.* (14) *can be swapped, i.e.,*

$$\max_{\boldsymbol{\beta}\in\mathcal{B}}\min_{\boldsymbol{\theta}\in\mathcal{R}^n} \mathcal{J}(\boldsymbol{\theta},\boldsymbol{\beta}) = \min_{\boldsymbol{\theta}\in\mathcal{R}^n}\max_{\boldsymbol{\beta}\in\mathcal{B}} \mathcal{J}(\boldsymbol{\theta},\boldsymbol{\beta}) \tag{16}$$

Proof. It can be verified that the following properties hold:

- $\mathcal{B}$ in Eq. (15) is a compact set under Euclidean distance.
- $\mathcal{J}(\cdot,\boldsymbol{\beta})$ is continuous and convex w.r.t. $\boldsymbol{\theta}$.
- $\mathcal{J}(\boldsymbol{\theta},\cdot)$ is continuous and concave (linear) w.r.t. $\boldsymbol{\beta}$.

Therefore, Sion's Minimax Theorem (Komiya 1988; Sion 1958) can be applied to Eq. (14) to swap the min and max operations. □

We denote $\mathcal{G}(\boldsymbol{\theta})$ as the objective function when $\boldsymbol{\beta}$ is set to its optimal value

$$\mathcal{G}(\boldsymbol{\theta}) = \max_{\boldsymbol{\beta}\in\mathcal{B}} \mathcal{J}(\boldsymbol{\theta}, \boldsymbol{\beta}) \tag{17}$$

Then we obtain the following objective which is equivalent to our min-max optimization problem (14)

$$\min_{\boldsymbol{\theta}\in\mathcal{R}^n} \mathcal{G}(\boldsymbol{\theta}) \tag{18}$$

Eqs. (10) and (12) show that $\mathcal{J}(\boldsymbol{\theta}, \boldsymbol{\beta})$ is a linear function w.r.t. $\boldsymbol{\beta}$ over convex set $\mathcal{B}$, and we formulate it as

$$\max_{\boldsymbol{\beta}\in\mathcal{B}} \boldsymbol{\beta}^T \boldsymbol{\phi} \tag{19}$$

where $\boldsymbol{\phi}$ denotes the coefficient of $\boldsymbol{\beta}$

$$\boldsymbol{\phi}_i = (\boldsymbol{K}_{i*}\boldsymbol{\theta})^2, i = 1, 2, ..., n_u \tag{20}$$

In the next few paragraphs, we show that the optimal solution of linear objective (19) can be easily obtained by just sorting all entries of $\boldsymbol{\phi}$. Therefore the objective function (19) is viewed as the sum of the largest n_s of entries. Additionally, this sum of entries is reformulated as point-wise maximum of several convex functions of $\boldsymbol{\theta}$. At this stage, we are pleased to find that $\mathcal{G}(\boldsymbol{\theta})$ is actually a convex function w.r.t. $\boldsymbol{\theta}$.

Lemma 2. *Sort $\{\phi_k\}_{k=1}^{n_u}$ in descend order as $\phi_{\pi_1} \geq \phi_{\pi_2} \geq ... \geq \phi_{\pi_{n_u}}$. Let $\hat{\boldsymbol{\beta}} \in \arg\max_{\boldsymbol{\beta}\in\mathcal{B}} \boldsymbol{\beta}^T\boldsymbol{\phi}$; then an optimal $\hat{\boldsymbol{\beta}}$ can be obtained by*

$$\hat{\beta}_i = \begin{cases} 1 & i \in \{\pi_1, \pi_2, ...\pi_{n_s}\} \\ 0 & otherwise \end{cases} \tag{21}$$

Proof. The problem of Eq. (19) can be reformulated as an $fractionalknapsackproblem$ (Cormen et al. 2001), where the objective is to select items with total weight of n_s from n_u items to obtain the largest total value. The i-th item has weight 1 and value ϕ_i, and items can be selected fractionally. β_i is the selected weight of the i-th item. The fractional knapsack problem can be solved by choosing the item with largest value-weight ratio iteratively (Cormen et al. 2001), so $\hat{\boldsymbol{\beta}}$ can be obtained by setting the entries corresponding to the largest n_s number of value ϕ_i to 1, and others to 0. □

Theorem 1. *$\mathcal{G}(\boldsymbol{\theta})$ is a convex function with respect to $\boldsymbol{\theta}$.*

Proof. We rewrite $\mathcal{G}(\boldsymbol{\theta})$ in the following form

$$\mathcal{G}(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{H}_0\boldsymbol{\theta} + \frac{1}{2}\lambda\boldsymbol{\theta}^T\boldsymbol{\theta} - \boldsymbol{b}^T\boldsymbol{\theta} + c_0 \max_{\boldsymbol{\beta}} \boldsymbol{\beta}^T\boldsymbol{\phi} \tag{22}$$

where $\boldsymbol{H}_0 = c_0\boldsymbol{K}_{XL}\boldsymbol{K}_{LX} + c_1\boldsymbol{K}^2$, λ, $\boldsymbol{b}$ and c_0 are constants.

The last term of Eq. (22) is a function of $\boldsymbol{\theta}$, and we denote it as $c_0\mathcal{F}(\boldsymbol{\theta})$. We further show that $\mathcal{F}(\boldsymbol{\theta})$ can be seen as point-wise maximum of convex functions with respect to to $\boldsymbol{\theta}$:

$$\mathcal{F}(\boldsymbol{\theta}) = \hat{\boldsymbol{\beta}}^T\boldsymbol{\phi} = \max_{\substack{\boldsymbol{a}\in\{0,1\}^{n_u} \\ \boldsymbol{a}^T\mathbf{1}=n_s}} \sum_{i=1}^{n_u} \phi_i a_i \tag{23}$$

In last step of the above equation, we used Lemma 2 and the simple fact that the sum of the largest n_s of entries is equivalent to the maximum sum of all combinations of n_s entries. From Eq. (20) we know that $\phi_i (i = 1, ...n_s)$ is convex with respect to $\boldsymbol{\theta}$, and $\sum_{i=1}^{n_u} \phi_i a_i$ is non-negative summation of ϕ_i. Additionally $\mathcal{F}(\boldsymbol{\theta})$ is a point-wise maximum of $\sum_{i=1}^{n_u} \phi_i a_i$. According to rules of convex functions, point-wise maximum and nonnegative summation preserve convexity, so $f(\boldsymbol{\theta})$ is a convex function with respect to $\boldsymbol{\theta}$.

It is easy to check that matrix $\boldsymbol{H}_0$ is semi-positive definite and $c_0 = (1-\alpha)/(n_s + n_l) > 0$, so $\mathcal{G}(\boldsymbol{\theta})$ is also a convex function w.r.t. $\boldsymbol{\theta}$. □

Now we have proved that the min-max objective $\mathcal{G}$ is a convex function. It is well known that convex functions always have $global$ optimal solutions, and there are many sophisticated algorithms to solve them.

Optimization

The subgradient method (Boyd, Xiao, and Mutapcic 2003) is shown to be a simple and effective technique to solve the non-differentiable convex programming problems without constraints (Boyd, Xiao, and Mutapcic 2003). It is also well-known that the subgradient method does not guarantee decreasing of the objective function at each iteration, but it lowers the Euclidean distance between the current point and the optimal point iteratively.

Since our objective function in Eq. (22) is a convex but non-differential function, we adopt the subgradient method to solve objective (18) directly. One of the prerequisite of the subgradient method is a subgradient of the given convex function. Here we compute a subgradient in Lemma 3.

Lemma 3. *In Eq. (22), a subgradient of $\mathcal{G}(\boldsymbol{\theta})$ is*

$$g(\boldsymbol{\theta}) = (\boldsymbol{H}_0 + \lambda\boldsymbol{I})\boldsymbol{\theta} - \boldsymbol{b} + 2c_0 \sum_{i=1}^{n_u} \hat{\beta}_i \boldsymbol{K}^{(i)}\boldsymbol{\theta} \tag{24}$$

Where $\boldsymbol{K}^{(i)} = \boldsymbol{K}_{*i}\boldsymbol{K}_{i*}$ and $\hat{\boldsymbol{\beta}}$ is obtained in Lemma 2.

Proof. We calculate $f(\boldsymbol{\theta})$ as a subgradient of $\mathcal{F}(\boldsymbol{\theta})$ by definition. Substituing ϕ_i into Eq. (23), we get

$$\mathcal{F}(\boldsymbol{\theta}) = \sum_{i=1}^{n_u} \hat{\beta}_i \boldsymbol{\theta}^T K^{(i)}\boldsymbol{\theta}$$

For all $\boldsymbol{x} \in \mathcal{R}^n$, we have

$$
\begin{aligned}
&\mathcal{F}(\boldsymbol{x}) - \mathcal{F}(\boldsymbol{\theta}) \\
&= \max_{\substack{\boldsymbol{a}\in\{0,1\}^{n_u} \\ \boldsymbol{a}^T\mathbf{1}=n_s}} \sum_{i=1}^{n_u} a_i \boldsymbol{x}^T K^{(i)} \boldsymbol{x} - \sum_{i=1}^{n_u} \hat{\beta}_i \boldsymbol{\theta}^T K^{(i)} \boldsymbol{\theta} \\
&\geq \sum_{i=1}^{n_u} \hat{\beta}_i (\boldsymbol{x}^T K^{(i)} \boldsymbol{x} - \boldsymbol{\theta}^T K^{(i)} \boldsymbol{\theta}) \\
&\geq (\boldsymbol{x} - \boldsymbol{\theta})^T \sum_{i=1}^{n_u} 2\hat{\beta}_i K^{(i)} \boldsymbol{\theta}
\end{aligned}
$$

The last step is obtained by expanding the inequality $(\boldsymbol{x} - \boldsymbol{\theta})^T K^{(i)} (\boldsymbol{x} - \boldsymbol{\theta}) \geq 0$ which holds for all $\boldsymbol{x} - \boldsymbol{\theta}$ because $K^{(i)}$ is symmetrical positive semi-definite. Therefore, $f(\boldsymbol{\theta}) = \sum_{i=1}^{n_u} 2\hat{\beta}_i K^{(i)} \boldsymbol{\theta}$ is the subgradient of function $\mathcal{F}(\boldsymbol{\theta})$. Taking derivatives of the rest terms of $\mathcal{G}(\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ and noticing $c_0 > 0$, the lemma is then proved according to the sum rule of subgradient. □

By denoting $\boldsymbol{\theta}^{(0)}$ as the starting point of the algorithm, and $\boldsymbol{\theta}^{(k)}$ as the point in the k-th iteration, we obtain the update rule

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - d_k g(\boldsymbol{\theta}^{(k)}) \tag{25}$$

where d_k is the step size. Since the objective does not guarantee to decrease in each iteration, a variable $\hat{\boldsymbol{\theta}}$ is introduced to obtain the best $\boldsymbol{\theta}$. Instead of a random initial point, we empirically choose the minimizer of $\frac{1}{2}\boldsymbol{\theta}^T \boldsymbol{H}_0 \boldsymbol{\theta} + \frac{1}{2}\lambda \boldsymbol{\theta}^T \boldsymbol{\theta} - \boldsymbol{b}^T \boldsymbol{\theta}$ as the starting point $\boldsymbol{\theta}^{(0)}$:

$$\boldsymbol{\theta}^{(0)} = (\boldsymbol{H}_0 + \lambda \boldsymbol{I})^{-1} \boldsymbol{b} \tag{26}$$

Algorithm 1 describes our proposed batch active learning approach RPE_{active} in detail. It can be seen from lines 5 to 7 that our algorithm runs in $\mathcal{O}((n_l + n_u)^2 n_s)$ each iteration.

Algorithm 1 Algorithm of RPE_{active}

Input: parameters α,λ; kernel matrix $\boldsymbol{K}$; constants n_u, n_l, n_s
Output: indicator variable $\boldsymbol{\beta}$

1: compute $\boldsymbol{\theta}^{(0)}$ according to (26)
2: $\hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^{(0)}$
3: $k \leftarrow 0$
4: **while** not converge **do**
5: compute $\hat{\boldsymbol{\beta}}$ according to (21)
6: compute $g(\boldsymbol{\theta}^{(k)})$ according to (24)
7: update $\boldsymbol{\theta}^{(k+1)}$ according to (25)
8: $k \leftarrow k + 1$
9: **if** $\mathcal{G}(\boldsymbol{\theta}^{(k)}) < \mathcal{G}(\hat{\boldsymbol{\theta}})$ **then**
10: $\hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^{(k)}$
11: **end if**
12: **end while**
13: compute $\boldsymbol{\phi}$ according to (20) with $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$
14: compute $\boldsymbol{\beta}$ according to (21)

Experimental Results

Experiment Setting

In our experiment, we evaluate the performance of our proposed RPE_{active} algorithm on 6 datasets from the UCI repository, namely iris, australian, sonar, heart, wine and arcene. Table 1 shows the detailed description of each dataset.

Table 1: Datasets Description

Dataset	#Instance	#Feature
iris	150	4
australian	690	14
sonar	208	60
heart	270	13
wine	178	13
arcene	100	10000

We compare our method to several state-of-art batch mode active learning algorithms. First, we consider the distribution-matching-based method using maximum mean discrepancy, denoted as $Mean$ (Chattopadhyay et al. 2013). It is most related to our method. Second, we compare with batch active learning based on transductive experimental design, denoted as $Design$ (Yu, Bi, and Tresp 2006). The third method is batch active learning using Fisher information matrix, denoted as $Fisher$ (Hoi et al. 2006). Finally, a baseline of random sampling is used and denoted as $Rand$.

We randomly divide each dataset into unlabeled set (60%) and testing set (40%). Each active learning algorithm selects data instances in the unlabeled set (60%) to query for labels and then the performance of each algorithm is measured by the classification accuracy on testing set (40%).

In our experiment, we mainly consider binary classification, and one instance of either class is randomly selected as the initial labeled data. All the algorithms start with the same initial, unlabeled and testing dataset. For a fixed batch size n_s, each method selects n_s data samples for labeling at each iteration. The batch size n_s is set to 5 in dateset iris and arcene due to their small sizes, and 10 in other datasets. The experiment is repeated 20 times and the average result is reported. Support Vector Machines is used as classification model to evaluate the performance of the labeled instances. Parameters α and λ are chosen from $\{0, 0.05, ..., 0.95\}$ and $\{10^{-5}, 10^{-4}, ..., 1\}$ respectively. We use Gaussian kernel for all datasets where the kernel width is searched in a relative

Table 2: The win/loss(%) of two-sided paired-t test in RPE_{active} vs $Fisher$, $Mean$, and $Design$ with $p < 0.05$

Dataset	vs $Fisher$		vs $Mean$		vs $Design$	
	Win	Loss	Win	Loss	Win	Loss
iris	78	0	33	0	78	0
australian	71	0	71	0	71	0
sonar	43	0	14	0	100	0
heart	0	0	9	0	9	0
wine	40	0	20	0	60	0
arcene	18	0	9	0	18	0

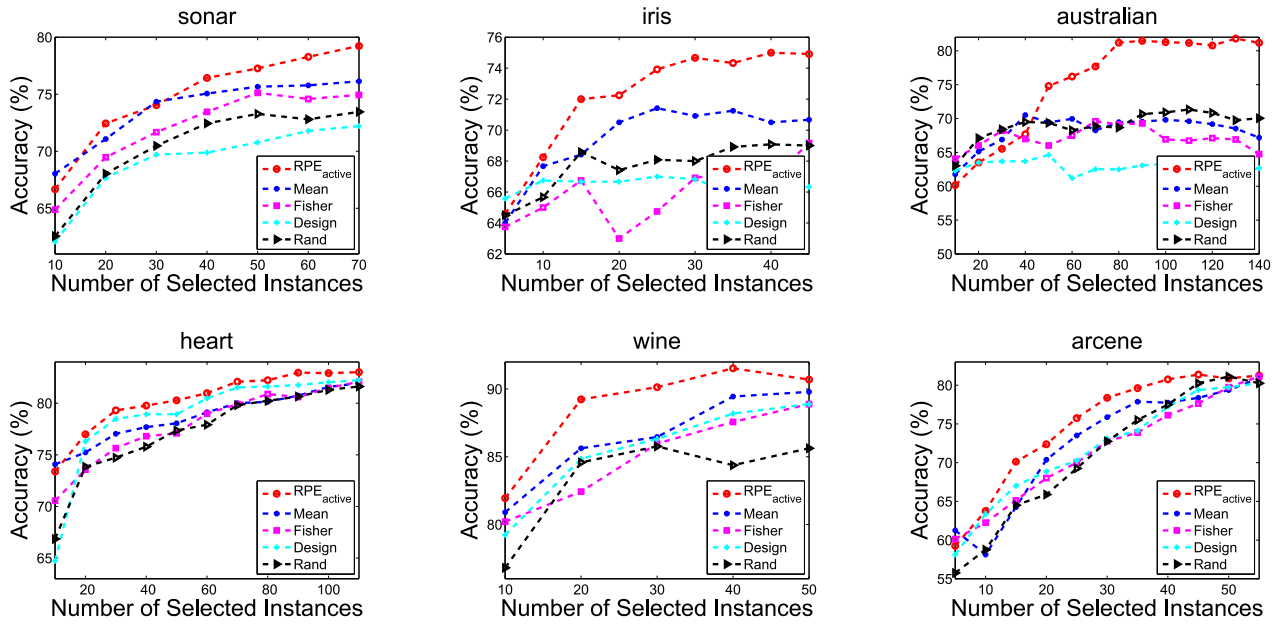

Figure 1: Average accuracy of RPE_{active}, $Mean$, $Fisher$, $Design$ and $Rand$ over 6 UCI datasets

large range. All the parameters are selected using 'greedy search' method which searches all combinations of parameters and the one with the best average accuracy on test data(40%) is chosen. The parameters are searched in each split of the total dataset. We use this scheme to all the methods in the experiment for fair comparison.

Comparative study

In this section, we conduct the two-sided paired t-test with $p < 0.05$ of our proposed method against $Fisher$, $Mean$ and $Design$ over all 6 datasets. Table 2 shows the performance of our method against the compared algorithms on each evaluation point of every dataset. The win/loss percentage shows the proportion of evaluation points that our method significantly outperforms/underperforms the compared algorithms. Figure 1 reveals the average accuracy over 20 runs on the 6 datasets. We can see that RPE_{active} outperforms all other methods on almost every evaluation points. Besides that, we can conclude from Table 2 that RPE_{active} outperforms other methods on all the datasets because RPE_{active} often significantly outperforms other algorithms and never loses in the t-test. Note that $Mean$ has slightly lower performance than our method in dataset heart and arcene, while $Design$ and $Fisher$ are the closest competitor in heart. Most importantly, our method has the best performance in australian, with a significantly large margin (71%) than other methods.

In the end, we compare RPE_{active} with $Mean$ because they are both based on distribution matching. From Figure 2 and Table 2, we find that RPE_{active} significantly performs $Mean$ on most datasets. On the other hand, $Mean$ never significantly outperforms our method although it only loses 9% in heart and arcene. Besides, all the results show that RPE_{active} outperforms $Mean$ when the labeled data becomes large, thus demonstrating the superiority of RPE over MMD in choosing the optimal batch.

Parameter Selection

The most important parameter of our proposed method is α. It is used to construct the relative distribution $q_\alpha(\boldsymbol{x}) = \alpha p(\boldsymbol{x}) + (1-\alpha)q(\boldsymbol{x})$, with $p(\boldsymbol{x})$ the distribution of D and $q(\boldsymbol{x})$ the distribution of $L \cup S$. We study the influence of α on classification accuracy by computing the average accuracy of RPE_{active} on two UCI datasets, namely heart and sonar, with α selected from $\{0.1, 0.5, 0.9\}$ and λ fixed to 10^{-5}. Figure 2 shows that the proposed RPE_{active} algorithm prefers the parameter α which is close to 1. Intuitively, our method minimizes the expected error on $q_\alpha(\boldsymbol{x})$ (see Eq. (2)). It is obvious that D is much larger than $L \cup S$ at the beginning and contains more information, so $q_\alpha(\boldsymbol{x})$ can be more precisely approximated with more proportions of the whole dataset.

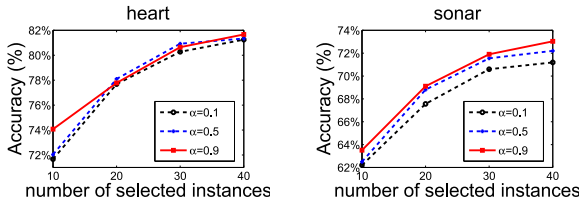

Figure 2: Average accuracy of RPE_{active} on *heart* and *sonar* with $\alpha = 0.1, 0.5, 0.9$.

Conclusion and Future Work

In this paper, we proposed a novel batch mode active sampling method that iteratively selects the batch of samples and minimizes the α-relative Pearson divergence (RPE) between the labeled and original data. We formulated the objective as a min-max optimization problem, and proved that it has an equivalent convex form. Then we applied a subgradient method to solve the convex objective. Our experiments on 6 UCI datasets demonstrated that our method significantly outperforms other state-of-the-art methods in general.

As future work, some problems of this framework remain unsolved. First, up to now, there is no theoretical solution to choose the direction of divergence estimation in active learning setting. In other words, we do not know how to choose between $\widehat{PE}_\alpha(L \cup S, D)$ and $\widehat{PE}_\alpha(D, L \cup S)$. Second, it is interesting to combine divergence estimation with other helpful information, such as label information, to further improve the performance. Third, due to that our method aims to find a global optimal solution, the computation is relatively slow to handle large datasets. It remains a challenge to develop faster algorithms to optimize the convex objective.

Acknowledgments

This work is supported in part by China National 973 program 2014CB340301 and NSFC grant 61379043.

References

Boyd, S., and Vandenberghe, L. 2009. *Convex optimization*. Cambridge university press.

Boyd, S.; Xiao, L.; and Mutapcic, A. 2003. Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter* 2004.

Campbell, C.; Cristianini, N.; Smola, A.; et al. 2000. Query learning with large margin classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 111–118.

Chattopadhyay, R.; Wang, Z.; Fan, W.; Davidson, I.; Panchanathan, S.; and Ye, J. 2013. Batch mode active sampling based on marginal probability distribution matching. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 7(3):13.

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Stein, C.; et al. 2001. *Introduction to Algorithms*, volume 2. MIT press Cambridge.

Dagan, I., and Engelson, S. P. 1995. Committee-based sampling for training probabilistic classifiers. In *Proceedings of the Twelfth International Conference on Machine Learning*, volume 95, 150–157. ACM.

Freund, Y.; Seung, H. S.; Shamir, E.; and Tishby, N. 1997. Selective sampling using the query by committee algorithm. *Machine learning* 28(2-3):133–168.

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.

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. ACM.

Komiya, H. 1988. Elementary proof for sion's minimax theorem. *Kodai mathematical journal* 11(1):5–7.

Kullback, S., and Leibler, R. A. 1951. On information and sufficiency. *The Annals of Mathematical Statistics* 79–86.

Schohn, G., and Cohn, D. 2000. Less is more: Active learning with support vector machines. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 839–846. ACM.

Settles, B., and Craven, M. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1070–1079. Association for Computational Linguistics.

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 fifth annual workshop on Computational learning theory*, 287–294. ACM.

Sion, M. 1958. On general minimax theorems. *Pacific J. Math* 8(1):171–176.

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. 2013. Querying discriminative and representative samples for batch mode active learning. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 158–166. ACM.

Yamada, M.; Suzuki, T.; Kanamori, T.; Hachiya, H.; and Sugiyama, M. 2011. Relative density-ratio estimation for robust distribution comparison. In *Advances in Neural Information Processing Systems*, 594–602.

Yu, K.; Bi, J.; and Tresp, V. 2006. Active learning via transductive experimental design. In *Proceedings of the 23rd International Conference on Machine Learning*, 1081–1088. ACM.