Batch Mode Active Sampling based on Marginal Probability Distribution Matching

Rita Chattopadhyay[1,2], Zheng Wang[1,2], Wei Fan[3], Ian Davidson[4], Sethuraman Panchanathan[1], Jieping Ye[1,2]
[1]Department of Computer Science and Engineering, Arizona State University, AZ 85287
[2]Center for Evolutionary Medicine and Informatics, The Biodesign Institute, Arizona State University, AZ 85287
[3]IBM T.J.Watson Research, Hawthorne, NY 10532
[4]Department of Computer Science, University of California, Davis, CA 95616

ABSTRACT

Active Learning is a machine learning and data mining technique that selects the most informative samples for labeling and uses them as training data; it is especially useful when there are large amount of unlabeled data and labeling them is expensive. Recently, batch-mode active learning, where a set of samples are selected concurrently for labeling, based on their collective merit, has attracted a lot of attention. The objective of batch-mode active learning is to select a set of informative samples so that a classifier learned on these samples has good generalization performance on the unlabeled data. Most of the existing batch-mode active learning methodologies try to achieve this by selecting samples based on varied criteria. In this paper we propose a novel criterion which achieves good generalization performance of a classifier by specifically selecting a set of query samples that minimizes the difference in distribution between the labeled and the unlabeled data, after annotation. We explicitly measure this difference based on all candidate subsets of the unlabeled data and select the best subset. The proposed objective is an NP-hard integer programming optimization problem. We provide two optimization techniques to solve this problem. In the first one, the problem is transformed into a convex quadratic programming problem and in the second method the problem is transformed into a linear programming problem. Our empirical studies using publicly available UCI datasets and a biomedical image dataset demonstrate the effectiveness of the proposed approach in comparison with the state-of-the-art batch-mode active learning methods. We also present two extensions of the proposed approach, which incorporate uncertainty of the predicted labels of the unlabeled data and transfer learning in the proposed formulation. Our empirical studies on UCI datasets show that incorporation of uncertainty information improves performance at later iterations while our studies on 20 Newsgroups dataset show that transfer learning improves the performance of the classifier during initial iterations.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications - Data Mining

General Terms

Algorithm

Keywords

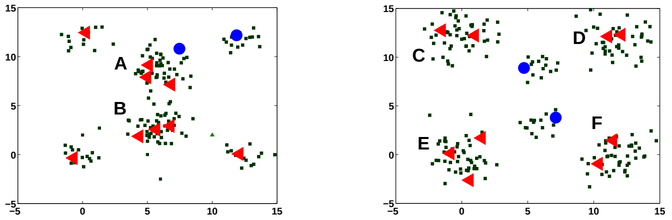
Active learning, marginal probability distribution, Maximum Mean Discrepancy

1. INTRODUCTION

Classification has been an active research topic in data mining and machine learning. The availability of a large amount of digital data in recent years, has greatly expanded the opportunities of automated data classification using data mining and machine learning techniques. One of the prerequisites for any data classification framework is the availability of labeled examples. Annotating large quantities of data for developing automated classifiers is a time consuming and expensive process. Hence there is a need to select an optimal set of instances from the pool of unlabeled data for labeling, such that a classifier learned on the selected instances performs well on the unlabeled data and also on unseen data belonging to the same distribution. Randomly selecting a set of unlabeled instances may result in selection of redundant and non-informative instances. *Active learning* methodologies enable selection of a set of most informative unlabeled instances from enormous amount of unlabeled data for manual labeling, with the intention of developing a good classifier with low generalization error. Specifically, the goal of active learning is to label as little data as possible, to achieve a certain classification performance, thus saving considerable annotation cost for training a good learner.

Active learning methodologies iteratively select the most informative data. Informativeness of a data sample or a set of data samples is measured by their potentiality in increasing the performance of a classifier, once their label is known [32]. Many researchers have addressed the active learning problem in various ways [23]. Most have focused on selecting a single most informative unlabeled instance to query each time. The most popular approaches include query-by-committee [24, 7, 9] where a number of distinct classification models are generated and an instance having the most disagreement among the classification models in predicting the label is selected for querying. Another popular approach is querying an instance with maximum uncertainty of labeling measured by the distance from the classification boundary [6, 22, 28] or by the entropy in the predicted label [17, 16].

Most single instance selection methods require to retrain the classifier with each single instance being labeled. The retraining process between queries can make the process very slow. Furthermore, if a parallel labeling system is available, e.g., multiple annotators working parallely, single instance selection would not be able to



(a) (b) (c)

Figure 1: Three toy data sets with different data distributions (dark green squares) and corresponding selected sets of query data points (red triangles) based on the proposed algorithm, selected in 3 iterations in batches of 3 data points. The two data points represented by blue circles are randomly selected initially available labeled data points (figures best viewed in color).

make the effective use of the resources. A batch-mode active learning strategy, that selects multiple instances each time is more appropriate under these circumstances. However, the challenge in batch-mode active learning is the formulation of the selection criteria for multiple instance selection. Using a single instance selection strategy to select a batch of queries in each iteration by ranking them based on their individual merit may not give good results, as this strategy fails to take into account the information overlap between multiple instances. Hence principles for batch-mode active learning to select a set of instances simultaneously based on their collective merit, need to be developed to address the multi-instance selection *specifically*.

The batch-mode active learning methods are particularly suitable for large scale applications where the data has high redundancy such as text classifications [11], content based image retrieval [14] and image recognition [13] due to high frame rate. The greatest challenge in selecting a set of instances simultaneously is two fold. The first challenge lies in the formulation of a right objective which will be optimized to select the most informative set of samples and the second challenge is concerned with the computational complexity of the NP hard combinatorial integer programming problem for obtaining a good local solution.

Recently, several sophisticated batch-mode active learning methods based on optimizing an information measure have been proposed for classification. Guo [11] selected a batch of query samples based on maximum mutual information with the unlabeled set of data, while Hoi et al. [13] applied Fisher information matrix to select a set of informative instances. Yu et al. [31] selected a set of instances closest to the basis vectors that approximate the set of unlabeled data and Guo and Schuurmans [12] proposed a discriminative approach.

In this paper, we propose a novel batch-mode active learning approach that selects a batch of query instances such that the distribution represented by the selected query set and the available labeled data is closest to the distribution represented by the unlabeled data. The motivation behind this approach is to ensure that a classifier learned by minimizing loss on the selected set of labeled data has good generalization capabilities on the unlabeled data and also on the unseen data coming from the same distribution. In other words, in order to learn a classifier with a budgeted number of labeled data we select a set of samples S from the unlabeled set of data, denoted by U, such that the probability distributions represented by $L \cup S$ and $U \backslash S$, where L is the set of available labeled data, are similar to each other.

We measure the difference in the probability distribution between the two sets of data using the Maximum Mean Discrepancy (MMD) proposed by Borgwardt et al [4, 1, 26]. Maximum Mean Discrepancy is a statistical test based on the fact that two distributions are different if and only if there exists at least one function in a characteristic RKHS [26] having different expectations on the two distributions. MMD has been proven to be very accurate in finding samples that were generated from the same distribution and outperforms its best competitors. MMD has been widely and successfully used in various classification tasks to ensure similarity in distribution between training and test data, specifically in the context of *transfer learning* applications [15, 20]. We use this measure in an optimization formulation to select a subset S out of all candidate subsets, based on minimum distribution difference between $L \cup S$ and $U \backslash S$. To the best of our knowledge, this is the first work that uses MMD in the active learning context.

Figure 1 shows the data points selected by the proposed method (red triangles) under three different distributions of unlabeled data (dark green squares). We created six dense regions of two different densities and kept a budget of nine query points which were selected in batches of three, in three iterations. We started with two randomly selected data samples (blue circles). We observe that the proposed method selects points from every dense region. It is also interesting to note that the number of points that get selected from each dense region is approximately proportional to the density of the region i.e., comparatively more data samples get selected from denser regions, A, B [Figure1 (a)], C, D, E, F [Figure1 (b)], and G, H [Figure1 (c)], thus preserving the distribution of the unlabeled data. We also observe that since available labeled data is considered at every iteration, hence diversity with respect to available data is maintained in query selection, as shown in Figures 1 (b) where no query data gets selected from the small regions in the center as an instance from those regions is already available in the initial labeled set. More details about the properties of query points are provided in Section 2.2. We also observe that the proposed approach decreases MMD monotonically as more data samples are selected from the unlabeled data and the decrease in MMD value corresponded to the increase in classification accuracy on the test set, discussed in detail in Section 4.

The subset selection problem is an NP-hard combinatorial integer programming problem. The proposed formulation is an integer quadratic programming problem. We show that the quadratic formulation can be reformulated as an integer linear programming problem. We then provide two optimization techniques to solve this problem. In the first method, we solve a continuous quadratic programming problem (by relaxing the integer constraint) on a convex function, providing a global solution. This is unlike most of the state-of-the art batch-mode active learning methods which provide a local solution following a gradient descent method [12, 11] or a greedy algorithm [13, 5, 30]. In the second method, we solve a continuous integer programming problem.

We tested our method on publicly available UCI[1] datasets and on a biomedical image dataset, Fly-FISH [18], consisting of images representing different developmental stages in the life cycle of *Drosophila*. The manual annotation of developmental stages of

[1] Available at http://www.ics.uci.edu/mlearn/MLRepository.html

the images of *Drosophila* is an expensive task. Hence active selection of an optimal number of images is crucial for the development of an automatic classifier. The empirical results on UCI and Fly-FISH datasets show that the proposed batch-mode active learning approach achieves superior or comparable performance to the state-of-the-art batch-mode active learning methods. The proposed method is also significantly time efficient compared to the state-of-the-art batch-mode active learning methods.

We further extend the proposed method by incorporating uncertainty of the predicted labels of the unlabeled data and transfer learning in the proposed formulation. Our empirical studies on UCI datasets show that incorporation of uncertainty information improves performance at later iterations while our studies on publicly available 20 Newsgroups dataset[2] show that transfer learning improves the performance of the classifier during initial iterations. The source codes along with synthetic data are available online (www.public.asu.edu/∼rchattop/code/MP-AL).

The remainder of the paper is organized as follows. Section 2 introduces the proposed batch-mode active learning framework. Section 3 compares the proposed method with the state-of-the-art active learning methods. Empirical studies are presented in Section 4. We present two extensions of the proposed approach in Section 5, and Section 6 concludes this paper.

2. PROPOSED FRAMEWORK

The key hypothesis in active learning is that if the learning algorithm is allowed to choose the data from which it learns, it will perform better even with less annotation [23]. Given a parametric classification model, the learning algorithms often learn the parameters θ by maximizing the joint probability $P(X,Y|\theta) = P(X|\theta)$ $P(Y|X,\theta)$ where X and Y are represented empirically by the training data $X_{tr} = \{x_1, x_2, \cdots, x_n\}$ and their corresponding labels $Y_{tr} = \{y_1, y_2, \cdots, y_n\}$ and $P(X)$ and $P(Y|X)$ denote the marginal and conditional probability distribution of X and Y respectively. Traditional data mining and machine learning algorithms are based on the assumption that the training data (X_{tr}, Y_{tr}) represents the true underlying distributions of X and Y and hence a model learned on this data works well for the test data (X_{tst}, Y_{tst}) which is also drawn i.i.d. from the same distribution. When the distributions on the training and test set are different the classification model learned on the training data performs poorly on the test data due to model mismatch.

The proposed active learning method addresses this issue by iteratively selecting a set of query instances from the unlabeled data such that the distribution represented by the queried and labeled data (X_{tr}, Y_{tr}), is similar to the probability distribution of the unlabeled data set. In other words, in order to learn a classifier with a budgeted number of labeled data, the proposed method iteratively selects a set of samples S from the unlabeled set of data, denoted by U, such that the joint probability distributions $P(X,Y)$ represented by $X_{tr} = L \cup S$ and $X_{tst} = U \backslash S$, where L is set of available labeled data, are similar to each other. Since the labeling function or the conditional probability $P(Y|X)$ remains the same for both S and $U \backslash S$ as they are drawn from the same underlying distribution, the problem reduces to selecting S such that the marginal probability $P_{S \cup L}(X)$ is similar to $P_{U \backslash S}(X)$. In this paper, we measure the difference in the marginal probability distribution between the two sets empirically using Maximum Mean Discrepancy (MMD) [4, 1, 26]. The difference between the empirical means of two distributions after mapping onto a reproducing kernel Hilbert space, called Maximum Mean Discrepancy, has been shown to be an effective measure of the difference in their marginal probability distributions. We review the basics of MMD below.

2.1 Maximum Mean Discrepancy (MMD)

Let $X = \{x_1, \cdots, x_m\}$ and $Z = \{z_1, \cdots, z_n\}$ be two sets of samples drawn randomly from a target population. Let p and q be the probability distributions defined on the basis of sample sets X and Z respectively. The Maximum Mean Discrepancy (MMD) proposed by Borgwardt et al [4, 1, 26] is a statistical tool that provides a method for testing whether two distributions p and q from which X and Z have been drawn respectively are similar or not.

The principal underlying the Maximum Mean Discrepancy is to find a function that assumes different expectations on two different distributions so that when evaluated empirically on samples drawn from the different distributions it would tell us whether the distributions are similar or not. Let $\mathcal{F}$ be a class of functions $f : \mathcal{X} \to \mathcal{R}$ and let X, Z, p, q be defined as above. Then the Maximum Mean Discrepancy and its empirical estimate are defined as:

$$\mathrm{MMD}[\mathcal{F}, p, q] \;:=\; \sup_{f \in \mathcal{F}} \left(E_p[f(x)] - E_q[f(z)] \right). \tag{1}$$

$$\mathrm{MMD}[\mathcal{F}, X, Z] \;:=\; \sup_{f \in \mathcal{F}} \left(\frac{1}{m} \sum_{i=1}^{m} f(x_i) - \frac{1}{n} \sum_{i=1}^{n} f(z_i) \right) \tag{2}$$

Intuitively if $\mathcal{F}$ is 'rich enough', $\mathrm{MMD}[\mathcal{F}, X, Z]$ will vanish if and only if $p = q$. A class of functions for which MMD may easily be computed, while retaining the ability to detect all discrepancies between p and q without making any simplifying assumptions is the complete inner product space $\mathcal{H}$ (i.e., a reproducing kernel Hilbert space (RKHS) [27]) of functions $f : \mathcal{X} \to \mathcal{R}$, where $\mathcal{X}$ is a nonempty compact set and for all $x \in \mathcal{X}$, the linear point evaluation functional mapping $f \to f(x)$ exists and is continuous. In this case, $f(x)$ can be expressed as an *inner product* via

$$f(x) = \langle \phi(x), f \rangle_{\mathcal{H}}. \tag{3}$$

where $\phi : \mathcal{X} \to \mathcal{H}$ is known as the feature space map from $\mathcal{X}$ to $\mathcal{H}$ [4]. When $\mathcal{F}$ is the unit ball in a characteristic RKHS [26], MMD is defined as the difference between the means of two distributions after mapping onto the characteristic RKHS. An empirical estimate of MMD is then obtained as follows:

$$\mathrm{MMD}[\phi, X, Z] := \left\| \frac{1}{m} \sum_{i=1}^{m} \Phi(x_i) - \frac{1}{n} \sum_{i=1}^{n} \Phi(z_i) \right\|_{\mathcal{H}}^{2}. \tag{4}$$

For more details about Maximum Mean Discrepancy, the related theoretical proofs and comparison with related methods, interested readers may refer to [4, 1, 26].

2.2 Proposed Batch Mode Active Sampling

The proposed batch-mode active sampling method, referred to as *Marginal Probability based Active Learning* (MP-AL) iteratively selects batches of query instances which represent best the distribution of the unlabeled instances so that a classifier learned by minimizing risk on the queried data after labeling, has good generalization performance on the unlabeled data set and on future unseen data that comes from the same distribution. We formulate the problem as an integer quadratic programming problem which can be reformulated as an equivalent integer linear programming problem.

The proposed framework uses MMD to measure the distribution difference between two sets of samples. Let us assume that we have n_u instances of unlabeled data U and n_l instances of labeled data L and we would like to select a batch S of b instances such that the distribution of $L \cup S$ is similar to the distribution of $U \backslash S$. In that

[2] Available at http://www.ai.mit.edu/∼jrennie/20Newsgroups/

case, the MMD between the sets $L \cup S$ and $U \backslash S$ defined by $f(S)$, can be computed using the expression in Equation (4), as follows:

$$f(S) = \left\| \frac{1}{n_l + b} \sum_{j \in L \cup S} \Phi(x_j) - \frac{1}{n_u - b} \sum_{i \in U \backslash S} \Phi(x_i) \right\|_{\mathcal{H}}^2 . \quad (5)$$

Since we want to select a set S that minimizes the mismatch between $L \cup S$ and $U \backslash S$ we propose to select a subset S of U that minimizes $f(S)$. Next we define a binary vector α of size n_u where each entry α_i indicates whether the data $x_i \in U$ is selected or not. If a point is selected, the corresponding entry α_i is 1 else 0. Thus the minimization problem reduces to finding α that minimizes the cost function $f(S)$:

$$\min_{\alpha:\alpha_i \in \{0,1\}, \alpha^T \mathbf{1} = b} \quad \left\| \frac{1}{n_l + b} \left(\sum_{j \in L} \Phi(x_j) + \sum_{i \in U} \alpha_i \Phi(x_i) \right) - \frac{1}{n_u - b} \sum_{i \in U} (1 - \alpha_i) \Phi(x_i) \right\|_{\mathcal{H}}^2 . \quad (6)$$

where $\mathbf{1}$ is a vector of the same dimension as α with all entries 1 and symbol T is used to represent the matrix or vector *transpose* operation. Evidently, the cost function in Equation (6) is an alternative (equivalent) representation of the cost function $f(S)$ in Equation (5). The first term denotes the mean of the mapped features of the labeled and selected points. Note that if a point x_i is not selected in the current set then α_i will be 0 and this term would not get added in the summation. The second term is mean of the mapped features of the unlabeled data set minus the selected query set. The first constraint ensures that each entry in α is either 0 or 1 and the second constraint ensures that exactly b entries of α are 1, meaning exactly b instances are selected from the unlabeled data set, where b is specified a priori by the user. The above formulation can be represented as:

$$\min_{\alpha:\alpha_i \in \{0,1\}, \alpha^T \mathbf{1} = b} \quad \frac{1}{2} \alpha^T K_1 \alpha - k_2^T \alpha + k_3^T \alpha + \text{const.} \quad (7)$$

The various terms in the above expression are given as follows. We denote G as the $(n_u + n_l) * (n_u + n_l)$ kernel Gram matrix over the unlabeled data U and labeled data L, arranged in order, using a kernel function K such that $G(i,j) = K(x_i, x_j)$. Then, $K_1 = G(1 : n_u, 1 : n_u)$, $k_2(i) = \frac{n_l + b}{n_l + n_u} \sum_{j=1}^{n_u} K_1(i,j)$, and $k_3(i) = \frac{n_u - b}{n_l + n_u} \sum_{j=1}^{n_l} G(i, n_u + j)$.

Based on the above expressions, we can draw the following observations regarding the properties of the selected query set:

- The first term ensures that the selected query set has minimum similarity within itself, avoiding *redundancy* in the selected set.
- The second term enforces the selected examples to be similar to the unselected ones, ensuring *representativeness*.
- The third term implies the examples with less similarity with already labeled data are more likely to be selected ensuring *diversity* in the selected set.

Thus the proposed method selects examples which meet all the desirable properties for batch mode active learning. The proposed method can be easily extended to add any other evaluation criteria (M) by adding a corresponding linear term $M^T \alpha$, while still maintaining the quadratic form (see Section 5 for more details). Also the proposed method does not depend on the availability of labeled data to initiate the process of selecting a query set, in which case $n_l = 0$, and the third term $k_3^T \alpha$ in Equation (7) vanishes.

The above optimization formulation is an integer quadratic programming problem. Next, we show that it can be reformulated as an equivalent integer linear programming (ILP) problem.

2.3 Reformulation as an ILP Problem

Due to the binary constraint $\alpha_i \in \{0,1\}$, $\forall i$, the linear terms in the objective defined in Equation (7) can be absorbed into the quadratic term by subtracting k_2 and adding k_3 terms to the diagonal entries of matrix K_1, forming a D matrix given by: $D(i,j) = K_1(i,j) - k_2(i) + k_3(i)$ for $i = j$ and $D(i,j) = K_1(i,j)$ otherwise. We can thus rewrite the optimization problem in (7) as:

$$\min_{\alpha:\alpha_i \in \{0,1\}, \alpha^T \mathbf{1} = b} \quad \alpha^T D \alpha. \quad (8)$$

We next introduce a binary matrix $Z = (z_{ij})$ with $z_{ij} = \alpha_i \alpha_j$. Thus the optimization in Equation (8) becomes:

$$\begin{aligned} \min_{\alpha, Z} \quad & \sum_{i,j} d_{ij} z_{ij} \\ \text{s.t.} \quad & z_{ij} = \alpha_i \alpha_j, \quad \alpha_i \in \{0,1\} \; \forall i,j, \quad \alpha^T \mathbf{1} = b. \end{aligned} \quad (9)$$

The quadratic equality constraint $z_{ij} = \alpha_i \alpha_j$ makes the problem difficult to solve. We show that this can be represented by a set of linear inequalities. Since d_{ij} can have both negative and positive values, we rewrite the quadratic constraints as follows:

$$\begin{aligned} \min_{\alpha, Z} \quad & \sum_{i,j} d_{ij} z_{ij} \\ \text{s.t.} \quad & -\alpha_i - \alpha_j + 2 z_{ij} \leq 0 \text{ for } d_{ij} < 0 \\ & \alpha_i + \alpha_j - z_{ij} \leq 1 \quad \text{for } d_{ij} \geq 0 \\ & \alpha_i \in \{0,1\} \; \forall i,j, \quad \alpha^T \mathbf{1} = b. \end{aligned} \quad (10)$$

The first constraint ensures that z_{ij} equals zero if the value of α_i or α_j (or both) is zero. If α_i and α_j both equal to one, the value of z_{ij} is free to be either 0 or 1; however, the minimization forces the value of z_{ij} to be 1 since d_{ij} is negative. Similarly, we derive the second constraint when d_{ij} is positive. The second constraint makes the value of z_{ij} equals to 1 if α_i and α_j both equal to one. If any of the α_i and α_j equals zero, then the value of z_{ij} is free to be either 0 or 1; nevertheless, at optimality, z_{ij} must equal 0 since d_{ij} has positive contribution to the objective, which is a minimization of the cost function. Consequently, the relation between z_{ij} and the pair α_i and α_j at optimality is as follows: $z_{i,j} = 1$, if and only if $\alpha_i = 1$ and $\alpha_j = 1$. Thus, the formulation in Equation (10) is equivalent to the original integer quadratic programming problem.

Next, we present two algorithms to solve the integer quadratic and integer linear optimization problems defined in Equation (7) and Equation (10) respectively as quadratic programming (QP) and linear programming (LP) problems by relaxing the binary constraints.

2.4 Quadratic Programming (QP) Problem

The binary constraint on α_i makes the integer quadratic problem defined in Equation (7) NP-hard. A common strategy is to relax the constraints to make it a continuous optimization problem, which can be solved in polynomial time:

$$\min_{\alpha: 0 \leq \alpha_i \leq 1, \alpha^T \mathbf{1} = b} \quad \frac{1}{2} \alpha^T K_1 \alpha - k_2^T \alpha + k_3^T \alpha. \quad (11)$$

The minimization problem in (11) is a standard quadratic problem (QP) and can be solved efficiently by applying many existing solvers. We used the ‘quadprog’ function in MATLAB to solve this QP problem. The overall algorithm for selecting the query set S at any iteration is given in Algorithm 1.

2.5 Linear Programming (LP) Problem

We relax the integral constraint and obtain a linear program (LP) formulation which is a relaxation of the ILP formulation in (10) as follows:

$$\begin{aligned} \min_{\alpha,Z} \quad & \sum_{i,j} d_{ij} z_{ij} \\ \text{s.t.} \quad & -\alpha_i - \alpha_j + 2z_{ij} \leq 0 \text{ for } d_{ij} < 0 \\ & \alpha_i + \alpha_j - z_{ij} \leq 1 \quad \text{ for } d_{ij} \geq 0 \\ & \alpha_i, z_{ij} \in [0,1] \ \forall i,j, \quad \alpha^T \mathbf{1} = b. \end{aligned} \tag{12}$$

The LP formulation can be further simplified by incorporating the first constraint into the objective function. Since this is a minimization problem when $d_{ij} < 0$, at optimality, $z_{ij} = \frac{\alpha_i+\alpha_j}{2}$ following the first equality constraint. On the other hand, the second equality constraint for $d_{ij} \geq 0$, may not hold at optimality. Since removing or relaxing a constraint of a minimization program does not reduce the optimal value, the formulation in Equation (12) can be reformulated as follows:

$$\begin{aligned} \min_{\alpha,Z} \quad & \frac{1}{2} \sum_{d_{ij}<0} d_{ij}(\alpha_i + \alpha_j) + \sum_{d_{ij} \geq 0} d_{ij} z_{ij} \\ \text{s.t.} \quad & \alpha_i + \alpha_j - z_{ij} \leq 1 \text{ for } d_{ij} \geq 0 \\ & z_{ij} \in [0,1] \text{ for } d_{ij} \geq 0 \\ & \alpha_i \in [0,1] \ \forall i,j, \quad \alpha^T \mathbf{1} = b. \end{aligned} \tag{13}$$

The problem in Equation (13) is a standard linear programming problem, and can be solved efficiently using any standard LP solver. Since the Hessian matrix K_1 in Equation (7) is a kernel Gram matrix which is positive semi-definite hence both formulations are convex. We used CVX [10] to solve the LP problem. The overall algorithm for selecting the query set S at any iteration using the LP formulation is given in Algorithm 1.

Algorithm 1 MP-AL

```
1: Input: $L$: set of labeled instances; $U$: set of unlabeled instances; $b$: batch size;
2: Output: $S$: query set;
3: Compute $K_1$, $k_2$ and $k_3$ as explained in Section 2.2.
4: if QP Problem then
5:     Compute $\alpha$ by solving (11).
6: end if
7: if LP Problem then
8:     Form $D$ matrix as explained in Section 2.3.
9:     Compute $\alpha$ by solving (13).
10: end if
11: Sort $U$ in descending order of $\alpha$ and select top $b$ instances as $S$.
12: Update sets $L$ and $U$: $L \rightarrow L \cup S$ , $U \rightarrow U \backslash S$.
```

3. COMPARISON WITH RELATED WORK

We compared the performance of the proposed method with state-of-the-art batch-mode active learning methods including *Matrix* [11], *Disc* [12] and *Fisher* [13] which selected a set of instances that are together maximally informative, similar to the proposed approach. We also compared our approach with state-of-the-art batch-mode active learning methods which selected a set of instances in each iteration based on their individual merits such as *svmD* [5] and a multiple criteria based instance selection method [30], referred to in this paper as *MCS* for convenience. Besides, we also compared our method to one transductive experimental design method [31], referred to as *Design*, which is based on regression models. A brief review of each of these methods is presented below.

The *Disc* method selects a set of instances by maximizing the likelihood of labeled and selected instances, while minimizing the uncertainty of unlabeled instances, based on a classifier learned on the labeled and selected instances. The problem formulation is non-convex and a local solution is obtained using the gradient descent method. The *Matrix* method selects a batch of queries S in each iteration by maximizing a mutual information criterion between the selected and labeled instances ($L' = L \cup S$) and unlabeled instances ($U' = U \backslash S$) as follows:

$$S^* = \arg \max_{|S|=b, S \subseteq U} ln|\Sigma_{L'L'}| + ln|\Sigma_{U'U'}|. \tag{14}$$

where $\Sigma_{U'U'}$ and $\Sigma_{L'L'}$ are covariance matrices of U' and L' computed using Gaussian kernel. Similar to the *Disc* method, this formulation is non-convex and a local solution is obtained using the gradient descent method. Similar to proposed approach, this method does not depend on any classifier model. The *Fisher* [13] method selects samples using Fisher information as the criterion. It selects a set of instances such that the difference in Fisher information between the selected set and unlabeled examples is minimum. The formulation is solved using a greedy algorithm.

The *svmD* method [5] selects a set of uncertain and diverse instances for query by ranking each instance in the unlabeled data based on their distance from the margin and maximum angle with the already labeled samples. The angle between two samples is measured using cosine of the angles between the hyperplanes corresponding to the samples. Similar to *svmD*, *MCS* [30] evaluates instances based on their individual merit using multi-criteria, but added a third term to measure the representativeness of each unlabeled data based on average cosine similarity with the unlabeled data. The proposed *MP-AL* method based on distribution matching, addresses both diversity and representativeness, besides addressing redundancy as well. It is also different from these two selection methods as it selects a batch of instances simultaneously which are together maximally informative based on their collective merit.

The *Design* method [31] proposes an experimental design in a *transductive* setting, where the focus is on the predictive performance on known test data. This method selects a set of instances closest to the basis vectors that approximate the set of unlabeled data. This method does not consider already labeled data, when selecting the next set of query, unlike the proposed method.

4. EXPERIMENTS AND RESULTS

Datasets. We evaluated the empirical performance of the proposed *MP-AL* algorithm using eight datasets from the UCI machine learning repository (both binary and multi-class), and a biological image dataset (Fly-FISH). The biological image dataset consists of 1016 images of 7 developmental stages of *Drosophila*, commonly known as fruit-fly. Each stage forms a class. Each image is represented by 3850 textural features that are extracted using Gabor filters [19].

Competing Methods. We compared the performance of the proposed approach with state-of-the-art batch-mode active learning methods which selected a set of instances based on their collective merit including *Matrix* [11], *Fisher* [13] and *Disc* [12]. We also compared our method with state-of-the-art batch-mode active learning methods which selected a set of instances based on their individual merit such as *svmD* [5] and *MCS* [30], besides comparing to one transductive experimental design method, referred to as *Design* [31]. We used the sequential design code downloaded from the authors' webpage and denote this method here as *Design*(s), that selects a set of instances sequentially based on their individual merits. A detailed description of each of these methods is provided in Section 3. Comparative performance of a random instance selection algorithm denoted as *Rand*, is also presented for reference.

Experimental Setup. We randomly divided each dataset into two sets. Batch selection based on active learning methodologies was performed on one set referred to as unlabeled set (65%) and the effectiveness of the selection methodologies was measured based on classification accuracy on the other unseen fixed set (35%) referred to as the test set. Table 2 summarizes the sizes of each of the datasets used. We subsampled some of the datasets due to the computational complexity of the several competing methods. We consider a hard case of active learning, where we start with two randomly selected labeled instances per class. All the algorithms start with the same initial labeled set, unlabeled set and test set. For a fixed batch size b, each algorithm repeatedly selected b instances for labeling at each iteration and evaluated the performance of a classifier learned on labeled instances, on the fixed test set. The size of b was fixed at 10 for all datasets except for Vehicles and Iris, where it was fixed at 5 due to their small sizes. The experiments were repeated 10 times and the average results are reported.

We compared the QP and LP formulations by the values of the objective function in Equation (7) and the classification accuracies obtained by the selected query set on the fixed test set. We observed that the values of the objective function obtained by LP were slightly lower than QP, though accuracies obtained were similar. We also noted that the execution time of QP was generally lower than LP. This can be attributed to the larger number of constraints in LP and the specific software package used. As part of the future work, we plan to explore ways to improve the efficiency of the LP formulation. The performance values of the proposed method included in this section are based on the QP formulation. We used a Gaussian kernel with the parameter value selected via cross validation and Support Vector Machines as classification model to evaluate the effectiveness of the queried instances.

Comparative Studies. The comparative performance of the proposed approach on UCI datasets, is shown in Figure 2. We observe that the proposed *MP-AL* performed better than the state-of-the-art batch-mode active learning methods for 6 out of 8 datasets and had comparable performance for the remaining two datasets: Musk and Wine. We also note that for 6 out of 8 datasets the nearest competitors were *Matrix*, *Fisher* and *Disc*, except for Iris and Vehicles where *svmD*, *MCS* and *Design*(s) were nearest competitors.

We conducted another set of experiments on a multi-class, high dimensional biological image dataset (Fly-FISH) for classifying different developmental stages of *Drosophila*. We randomly sampled 511 samples (divided equally among all 7 classes). The batch size b and number of iterations were fixed at 10 and 9 respectively. Figure 4 reports the results of the proposed method *MP-AL* compared to the other active learning methods. We observe that *MP-AL* outperformed all the other active learning methods followed by *Matrix*, *Fisher* and *Disc*. Table 1 presents the results of 2-sided paired t-test of *MP-AL* vs *Matrix*, *Disc* and *Fisher* methods on UCI and Fly-FISH datasets. We compare the accuracies over 10 runs at each evaluation point and present the percentage of evaluation points at which *MP-AL significantly* outperforms or under-performs the compared algorithm, denoted as win % and loss% respectively.

Variation in MMD Vs Number of Selected Samples. We investigated the variation in MMD value between the training set and the unlabeled data at each iteration for all the datasets. Figure 3 presents the results for some of the representative UCI datasets. Similar patterns were observed for the other datasets. We observe that our algorithm decreases MMD value monotonically as more data samples are selected from the unlabeled data and that the decrease in MMD value corresponds to the increase in classification accuracy on the test data as shown in Figure 2. The decrease in MMD value during the initial iterations is more than the decrease towards the later iterations, resulting in the higher increase in accuracy values during the initial iterations than later iterations. We observe for the Vehicles dataset the accuracy value sharply increases between the second and third iteration points. We also observe a sharp decrease in MMD value for the Vehicles dataset at the corresponding iteration points.

Table 1: Win-Loss % of MP-AL in 2-sided paired t-test ($p <$ 0.05). The fraction not reported (e.g., 60% for MP-AL vs. Matrix on Heart) corresponds to the cases where the two algorithms are not significantly different at the level of $p < 0.05$.

Dataset	MP-AL vs. Matrix		MP-AL vs. Disc		MP-AL vs. Fisher	
	win%	lose%	win%	lose%	win%	lose%
Heart	40.0	0	60.0	0	60.0	0
Sonar	28.5	0	85.7	0	28.5	0
Waveform	14.3	0	71.4	0	28.5	0
Image Segmentation	20.0	0	60.0	0	40.0	0
Musk	0	20.0	0.0	0	20.0	0
Wine	0.0	0	75.0	0	50.0	0
Vehicles	66.6	0	50.0	0	83.3	0
Iris	80.0	0	80.0	0	100.0	0
Fly-FISH	62.5	0	85.7	0	62.5	0

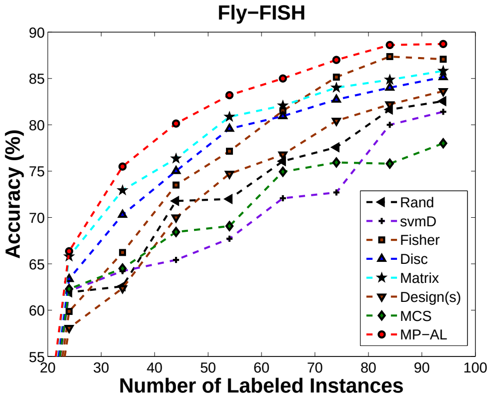


Figure 4: Comparative performance of different active learning methods on the Fly-FISH dataset.

Table 2: Average run time (in seconds).

Dataset	# instances x features	MP-AL	Matrix	Disc	Fisher
Vehicles	94 x 19	0.42	3.35	4.42	1.14
Iris	150 x 5	0.58	6.64	7.97	2.96
Wine	178 x 3	0.59	7.81	10.76	8.38
Sonar	208 x 60	0.76	8.68	12.78	10.04
Image Segmentation	210 x 19	0.76	12.47	22.43	16.51
Heart	270 x 13	1.21	26.14	26.93	18.88
Waveform	350 x 21	2.54	52.21	81.28	38.57
Musk	476 x 167	3.81	153.02	265.71	121.23
Fly-FISH	511 x 3850	6.5	360.70	3330.16	923.30

Efficiency Comparison. We compared the average time taken to select a batch of unlabeled points by the proposed *MP-AL* versus the nearest competitors *Matrix*, *Fisher* and *Disc*. All algorithms were implemented using MATLAB on a four-core Intel processor with 2.66 GHz CPU and 8 GB RAM. Table 2 presents the comparative run times on different UCI and Fly-FISH datasets. We note that *MP-AL* is much more efficient than the other three batch-mode active learning methods for all datasets. *Matrix* method involved solving a quadratic programming problem multiple times, per batch of query points selection. *Fisher* involved training of a classifier multiple times per query batch selection. The *Disc* method involved training of a classifier followed by solving a quadratic programming problem multiple times per selection of a query batch and the *MP-AL* on the other hand, required solving a quadratic programming problem once per batch of query points selection.

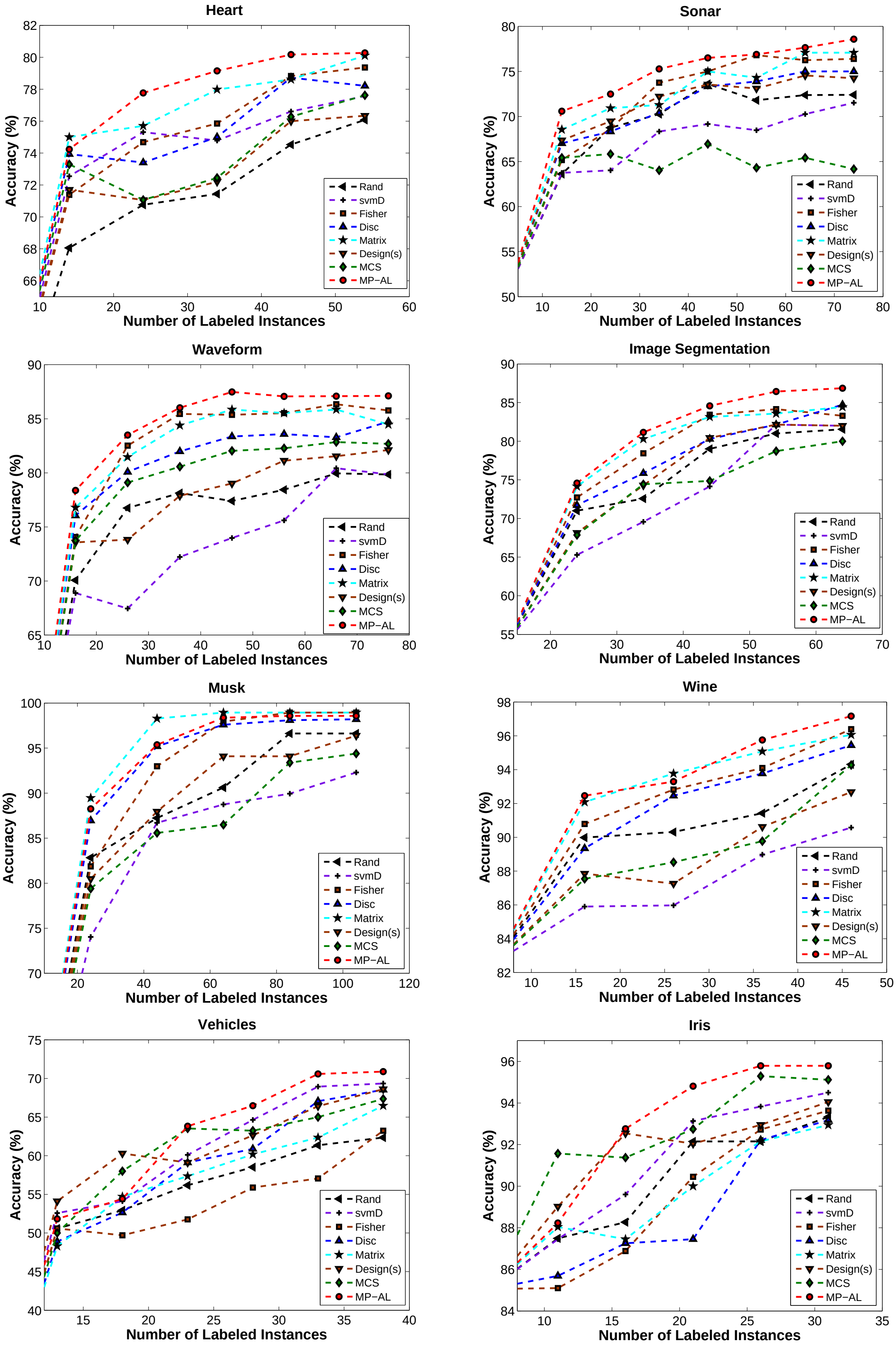


Figure 2: Comparative performance of different active learning methods on UCI datasets. Accuracy at the start point, which is same for all methods is not shown in the figures (figures best viewed in color). Results of 2-sided paired t-test at the level of $p < 0.05$ for *MP-AL* vs. *Matrix*, *Fisher* and *Disc* are presented in Table 1.

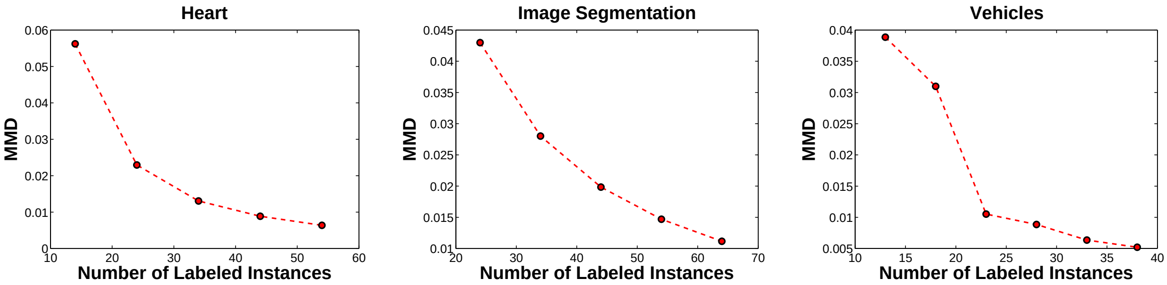


Figure 3: MMD value between the training and unlabeled data as more instances are selected by MP-AL.

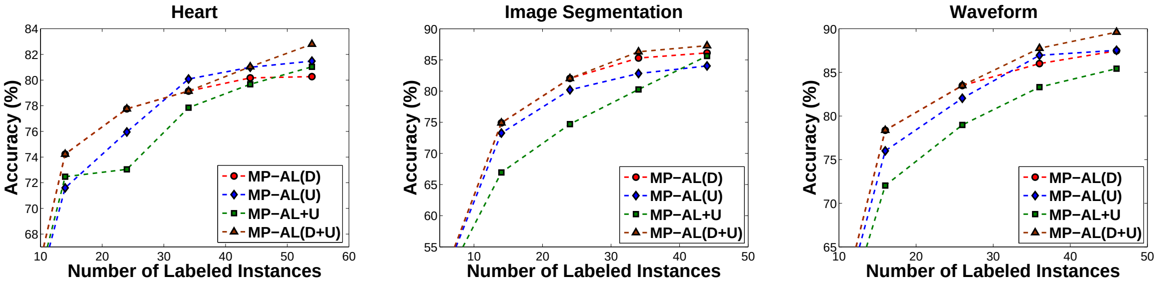


Figure 5: MP-AL with Uncertainty on UCI datasets.

5. EXTENSIONS OF MP-AL

5.1 Incorporating Uncertainty in MP-AL

The proposed *MP-AL* framework can be readily extended to incorporate uncertainty of prediction of unlabeled data in the query selection process. We experimented with the following three alternative algorithms: (1) At each iteration a classifier is learned on the available labeled data, and the unlabeled data is ranked based on uncertainty of predictions $M(x_i)$ measured for each unlabeled data x_i using entropy of predicted labels as in [12]. The *MP-AL* method is then applied only on the most uncertain set of unlabeled data (top 70%) instead of on the complete unlabeled data, referred to as *MP-AL*(U). (2) Query selection is based on *MP-AL* at initial iterations and during later iterations based on the *MP-AL*(U), referred to as *MP-AL*(D+U). (3) Add the prediction uncertainty vector M as a separate linear term $(-M^T\alpha)$ in the formulation in Equation (11), referred to as *MP-AL+U*. The base *MP-AL* method is referred to as *MP-AL*(D). Figure 5 presents the comparative results obtained on some of the representative UCI datasets. Similar patterns were obtained on other datasets. We observe that for initial iterations *MP-AL*(D) and *MP-AL*(D+U) outperform *MP-AL*(U) and *MP-AL+U*. However the performance of these methods improve during later iterations, as the classifier becomes more reliable when learned on a larger number of labeled data. Indeed we can observe that *MP-AL*(D+U) performs best for most cases as it combines the strengths of both *MP-AL*(D) and *MP-AL*(U).

5.2 Transfer Learning in MP-AL

The problem of insufficient labeled data (in a target domain) is addressed in *Active learning* by querying labels of most informative instances. An alternative to address the same problem is by borrowing samples from an already labeled dataset belonging to a related domain (source domain), known as *Transfer Learning* [21]. Different transfer learning methodologies have been developed to address the distribution difference between a source and a target domain, so that the source domain data can be efficiently used to label the target domain data. Re-weighting source domain data to match the marginal probability distributions is a commonly used strategy [15, 3, 25] in transfer learning. In our experiments we used an existing re-weighting method [15], to re-weight the source domain data to match the distribution of the unlabeled set (target domain). We incorporate transfer learning in our MP-AL framework as follows: At each iteration, we combine the re-weighted source samples with the queried and labeled samples from the unlabeled set (in the target domain) and compute the classification accuracy on the other unseen fixed test set, similar to earlier experiments. We evaluated the proposed transfer learning extension of *MP-AL* on the 20 Newsgroups dataset for document categorization. We built three sets of source domain data vs. unlabeled data (target domain) as follows: (1) Sports: rec.sport.hockey vs. rec.sport.baseball; (2) Hardware: comp.sys.mac.hardware vs. comp.sys.ibm.pc.hardware and (3) Scientific: sci.med vs. sci.electronics. The positive class of each source and target domain data consists of 200 documents randomly sampled from the respective categories and the negative class consists of a random mixture of 200 samples from other categories as suggested in [8]. We represented each document as a binary vector consisting of the 200 most discriminating words determined by Weka's info-gain filter [29], after removing stop words and using a document frequency of 5. We start with no selected labeled instances and use a batch size of 10. The experiments were repeated 10 times and the average results are reported. Figure 6 shows the accuracies obtained by the extended method denoted as *MP-AL+TL* and by the base *MP-AL* method without using the source domain data, on the 20 Newsgroups dataset. We observe that during intial iterations *MP-AL* has poorer performance, due to insufficient labeled data. We observe that combining transfer learning with active learning (*MP-AL+TL*) improves the classification accuracy significantly during the initial iterations. However as the number of labeled data from the unlabeled set (target domain) increases, the performance of *MP-AL* improves and outperforms *MP-AL+TL* for Scientific and Hardware test cases. It has been shown both theoretically and empirically in [2] that if there are enough data from the target domain then no source data are needed, and in fact using additional source data may degrade the performance. We also observe that improvement in classification accuracies due to incorporation of transfer learning is more for Sports and moderate for Scientific and Hardware. This can be attributed to the extent of difference in distribution between the source and target domains

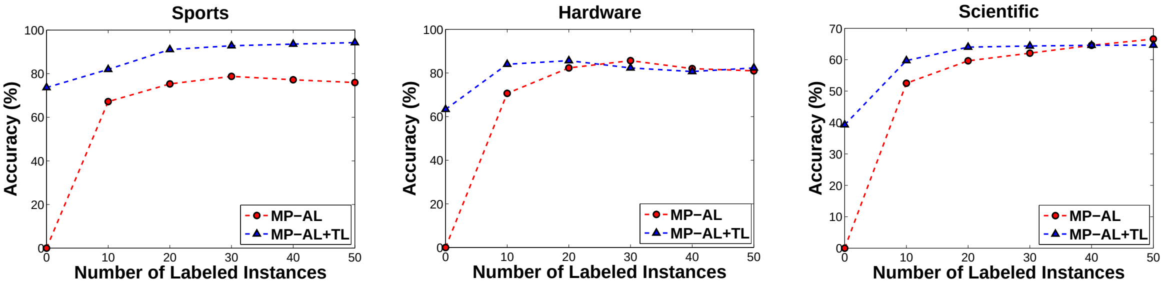


Figure 6: MP-AL with transfer learning on 20 Newsgroups dataset.

in each of these test cases; one way to measure the distribution difference is to compute the MMD value between the source and target domains. The MMD value is 0.0121, 0.0237 and 0.0239 for Sports, Hardware and Scientific respectively. This is consistent with our observation in Figure 6.

6. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel batch-mode active learning method that selects a set of query samples from the unlabeled data so that the marginal probability distribution represented by the labeled data after annotation, is similar to the marginal probability distribution represented by the unlabeled data. The motivation behind this approach is to ensure that a classifier learned on labeled data with similar distribution has good generalization performance on the unlabeled data and also on the unseen data coming from the same distribution. The proposed method fully explores the available unlabeled and the already labeled data and demonstrates sensible data selection properties. It is formulated as an integer quadratic programming problem; besides, we also present an equivalent linear programming formulation. Our empirical studies show that the proposed approach achieves superior or comparable performance, besides being computationally highly efficient, compared to the existing batch-mode active learning methods. In addition, we present two extensions by incorporating uncertainty of the predicted labels of the unlabeled data and transfer learning in the proposed formulation. Our empirical studies on UCI and 20 Newsgroup datasets show that incorporating uncertainty information in the proposed formulation improves performance at later iterations and including transfer learning improves the performance of the classifier during initial iterations. In future work, we plan to study the theoretical properties of the proposed formulation. In addition, we plan to extend MP-AL to the multi-label setting.

7. ACKNOWLEDGMENTS

The authors are grateful to Prof. Arthur Gretton (Gatsby Computational Neuroscience Unit, UCL, London, UK) for providing valuable suggestions and feedback. This research is sponsored in part by NSF IIS-0953662, CCF-1025177, NIH LM010730, and ONR N00014-11-1-0108.

8. REFERENCES

[1] A.Gretton, K.M.Borgwardt, M.Rasch, B.Schölkopf, and A.J.Smola. A kernel method for the two-sample-problem. In *NIPS*, 2007.

[2] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Vaughan. A theory of learning from different domains. *JMLR*, 79:151–175, 2010.

[3] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *JMLR*, 10:2137–2155, 2009.

[4] K. Borgwardt, A. Gretton, M. Rasch, H. Kriegel, B.Schölkopf, and A. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):49–57, 2006.

[5] K. Brinker. Incorporating diversity in active learning with support vector machines. In *ICML*, 2003.

[6] C. Campbell, N. Cristianini, and A. Smola. Query learning with large margin classifiers. In *ICML*, 2000.

[7] I. Dagan and S. Engelson. Committee-based sampling for training probabilistic classifiers. In *ICML*, 1995.

[8] E. Eaton and M. desJardins. Set-based boosting for instance-level transfer. In *ICDM*, 2009.

[9] Y. Freund, H. Seung, E. Shamir, and N. Tishby. Selective sampling using query by committee algorithm. *Mach. Learn.*, 28, 1997.

[10] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21, 2007.

[11] Y. Guo. Active instance sampling via matrix partition. In *NIPS*, 2010.

[12] Y. Guo and D. Schuurmans. Discriminative batch mode active learning. In *NIPS*, 2007.

[13] S. Hoi, R. Jin, J. Zhu, and M. Lyu. Batch mode active learning and its application to medical image classification. In *ICML*, 2006.

[14] S. Hoi, R. Jin, J. Zhu, and M. Lyu. Semi-supervised svm batch mode active learning for image retrieval. In *CVPR*, 2008.

[15] J. Huang, A. Smola, A. Gretton, K. Borgwardt, and B. Scholkopf. Correcting sample selection bias by unlabeled data. In *NIPS*, 2007.

[16] F. Jing, M. Li, H. Zhang, and B. Zhang. Entropy based active learning with support vector machines for content based image retrieval. In *ICME*, 2004.

[17] A. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *CVPR*, 2009.

[18] E. Lecuyer, H. Yoshida, N. Parthasarathy, C. Alm, and T. Babak. Global analysis of mrna localization reveals a prominent role in organizing cellular architecture and function. *Cell*, 2011.

[19] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Trans. Image Processing*, 11:467–476, 2002.

[20] S. Pan, I. Tsang, J. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. In *IJCAI*, 2009.

[21] S. Pan and Q. Yang. A survey on transfer learning. *TKDE*, 2009.

[22] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *ICML*, 2000.

[23] B. Settles. Active learning literature survey. In *Computer Sciences Technical Report 1648*. University of Wisconsin-Madison, 2009.

[24] H. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *ACM Workshop on Computational Learning Theory*, 1992.

[25] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *JSPI*, 90:227–244, 2000.

[26] B. Sriperumbudur, A. Gretton, K. Fukumizu, B.Schölkopf, and G. Lanckriet. Hilbert space embeddings and metrics on probability measures. *JMLR*, 11:1517–1561, 2010.

[27] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *JMLR*, 2:67–93, 2002.

[28] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *JMLR*, 2:45–66, 2000.

[29] I. Witten and E. Frank. Data mining: Practical machine learning tools with java implementations. Morgan Kaufmann, 2000.

[30] Y. Wu, I. Kozintsev, J. Bouguet, and C. Dulong. Sampling strategies for active learning in personal photo retrieval. In *ICME*, 2006.

[31] K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In *ICML*, 2006.

[32] T. Zhang and F. Oles. A probability analysis on the value of unlabeled data for classification problems. In *ICML*, 2000.