Active Batch Selection via Convex Relaxations with Guaranteed Solution Bounds

Shayok Chakraborty, *Member, IEEE*, Vineeth Balasubramanian, *Member, IEEE*, Qian Sun, *Member, IEEE*, Sethuraman Panchanathan, *Fellow, IEEE*, and Jieping Ye, *Senior Member, IEEE*

Abstract—Active learning techniques have gained popularity to reduce human effort in labeling data instances for inducing a classifier. When faced with large amounts of unlabeled data, such algorithms automatically identify the exemplar instances for manual annotation. More recently, there have been attempts towards a batch mode form of active learning, where a batch of data points is simultaneously selected from an unlabeled set. In this paper, we propose two novel batch mode active learning (BMAL) algorithms: BatchRank and BatchRand. We first formulate the batch selection task as an NP-hard optimization problem; we then propose two convex relaxations, one based on linear programming and the other based on semi-definite programming to solve the batch selection problem. Finally, a deterministic bound is derived on the solution quality for the first relaxation and a probabilistic bound for the second. To the best of our knowledge, this is the first research effort to derive mathematical guarantees on the solution quality of the BMAL problem. Our extensive empirical studies on 15 binary, multi-class and multi-label challenging datasets corroborate that the proposed algorithms perform at par with the state-of-the-art techniques, deliver high quality solutions and are robust to real-world issues like label noise and class imbalance.

Index Terms—Batch mode active learning, optimization

1 Introduction

The advance of technology in leaps and bounds, together with the widespread emergence of modern technological equipments has resulted in the generation of humongous amounts of digital data (in the form of images, audio and text among others) in the modern era. While this has expanded the possibilities of solving real world problems using computational learning frameworks, selecting the salient samples from such a huge collection of data has proved to be a significant and useful challenge. Further, to train a reliable classification model, it is indispensable to have a large quantity of labeled training data. Manual annotation of such a large scale data is an expensive process in terms of time, labor and human expertise. This has set the stage for research in the field of active learning. Active learning algorithms automatically select the salient and exemplar instances from large quantities of unlabeled data and thereby tremendously reduce human annotation effort in training an effective classifier.

With the advent of technologies like the Amazon Mechanical Turk, it is now possible to leverage the intelligence of multiple human users simultaneously in labeling data instances to train a classification model. To this end, *batch mode active learning* (BMAL) techniques have been proposed in recent years. Such algorithms attempt to select a batch of unlabeled data points simultaneously from an unlabeled set instead of a single instance at a time. Sample applications of such a scheme include image retrieval [1], medical image classification [2] and text mining [3].

State-of-the-art BMAL techniques typically formulate the batch selection task as an NP-hard integer programming problem and convex relaxations of the NP-hard problems are then solved to select an appropriate batch of unlabeled samples [4], [5]. Even though such techniques depict promising empirical performance, no formal mathematical guarantee has been established on the solution qualities of the convex relaxations. In this work, we first formulate the batch selection as an NP-hard integer quadratic programming (IQP) problem. We then propose a linear programming (LP) based relaxation and show that the batch selection task reduces to a score ranking problem; we hence call this method *BatchRank*. We also propose a semi-definite programming (SDP) based relaxation and use randomization techniques to solve the BMAL problem; we therefore name this algorithm *BatchRand*. Further, we derive a deterministic bound on the quality of the first relaxation and a probabilistic bound on the quality of the second. To the best of our knowledge, this is the first research effort to provide concrete mathematical guarantees on the solution quality of the batch mode active learning problem. We note here that the purpose of this work is not to derive a bound on the number of queries required to achieve a given generalization error in active learning.

- *S. Chakraborty is with the Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh, PA 15213. E-mail: shayokc@andrew.cmu.edu.*
- *V. Balasubramanian is with the Computer Science and Engineering Department, Indian Institute of Technology, Hyderabad, India. E-mail: vineethnb@iith.ac.in.*
- *Q. Sun and J. Ye are with the Computer Science and Engineering Department, Arizona State University, Tempe, AZ 85281. E-mail: {qian.sun, jieping.ye}@asu.edu.*
- *S. Panchanathan is with the Center for Cognitive Ubiquitous Computing (CUbiC), Arizona State University, Tempe, AZ 85281. E-mail: panch@asu.edu.*

Manuscript received 25 Apr. 2013; revised 1 Oct. 2014; accepted 29 Dec. 2014. Date of publication 11 Jan. 2015; date of current version 4 Sept. 2015. Recommended for acceptance by G. Lanckriet.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2389848

This problem has been extensively studied in the literature [6], [7], [8]. Our objective is to derive mathematical guarantees on the solution qualities of the convex relaxations of the batch mode active learning problem, which, to the best of our knowledge, has not been addressed till date. We also empirically validate that the proposed algorithms deliver high quality solutions and are robust to label noise and class imbalance.

The rest of the paper is organized as follows: we present a survey of existing active learning techniques in Section 2, we then detail the generic batch mode active learning framework in Section 3 and derive the two convex relaxations in Sections 4 and 5, the results of our experiments are presented in Section 6 and we conclude with discussions in Section 7.

2 Related Work

Active learning is a well-studied problem in machine learning literature. Several techniques have been developed over the last few years and a review of these can be found in [9]. All these techniques have been developed within the scope of two distinct settings—*online* and *pool-based*. In online active learning, the learner encounters the data points sequentially and at each instant, it needs to decide whether the current point needs to be queried for its class label [10], [11], [12]. In contrast, in a pool based setting, the learner is exposed to a pool of unlabeled data instances and it iteratively selects points for manual annotation. Pool based active learning is further classified into *single instance selection* and *batch selection*.

In single instance selection, the learner selects the single most informative unlabeled instance to query each time. Such techniques can be broadly categorized into four groups—(i) SVM based approaches, which decide the next point to be queried based on its distance from the hyperplane in the feature space [13], [14], (ii) Statistical approaches, which query points such that some statistical property of the future learner (e.g., the learner variance) is optimized [15], (iii) Query by Committee, which chooses points to be queried based on the level of disagreement among an ensemble of classifiers [16], [17] and (iv) Information theoretic approaches, which exploit the discriminative partition information contained in the unlabeled data and query the instance that provides the maximum conditional mutual information about the labels of the unlabeled instances, given the labeled data, in an optimistic way [18]. In other kinds of similar approaches, Baram et al. [19] proposed a master algorithm which estimated the progress of each active learner and dynamically switched over to the best performing one at the current stage. McCallum and Nigam [20] combined the Expectation Maximization algorithm with naive Bayes classifier to learn from labeled and unlabeled text documents. Recent efforts in this area have included a novel min-max approach to systematically combine multiple criteria (such as informativeness and representativeness) for active sample selection [21], an importance-weighting based practical approach which has guaranteed label complexity bounds [6], a framework for networked data which exploited the links between instances and the interaction between the local and collective aspects of a classifier for point selection [22], as well as application of active learning approaches to rapidly improve a multi-task adaptive filtering system with minimal user/task-level feedback [23].

To avoid frequent classifier retraining and to utilize the presence of parallel labeling oracles, *batch mode active learning* schemes, which select multiple unlabeled points simultaneously for manual annotation, have been proposed in recent years. Existing approaches for batch mode active learning have largely been based on extending pool-based active learning methods to select multiple instances simultaneously. They use greedy heuristics and select the top k informative instances (k being the required batch size) from the unlabeled set for manual annotation. Brinker [24] extended the version space concept proposed in [13] to query a diverse batch of points using SVMs, where diversity was measured as the angle induced by the hyperplane of the currently selected point to the hyperplanes of the already selected points. Schohn and Cohn [25] proposed to query a batch of points based on their distance from the separating hyperplane of a linear SVM. Xu et al. [26] proposed an SVM based BMAL strategy which combined representativeness and diversity measures for batch selection.

However, extending the pool-based setting to the batch setting by considering the top k instances does not account for other factors such as information overlap among the selected points in a batch. More recently, this has led to newer efforts that are specifically intended to select batches of points using appropriate optimization strategies. Hoi et al. [1], [3] used the Fisher information matrix as a measure of model uncertainty and proposed to query the set of points that maximally reduced the Fisher information. The same authors [27] proposed a BMAL scheme based on SVMs where a kernel function was first learned from a mixture of labeled and unlabeled samples, which was then used to identify the informative and diverse examples through a min-max framework. Chakraborty et al. [28] proposed a generalized BMAL scheme, based on Quasi-Newton optimization, and applied it to the face-based biometric recognition problem. The same framework was also shown to address active learning from multiple sources of information and context aware active sample selection. Guo and Schuurmans [4] proposed a discriminative strategy that selected a batch of points which maximized the log-likelihoods of the selected points with respect to their optimistically assigned class labels and minimized the entropy of the unselected points in the unlabeled pool. Very recently, Guo [5] proposed a batch mode active learning scheme which maximized the mutual information between the labeled and unlabeled sets and was independent of the classification model used. The methods described in [4] and [5] have been shown to be the best performing BMAL schemes till date [5].

While these algorithms empirically outperformed the existing heuristic schemes in terms of classification accuracy, no theoretical guarantee was established on the quality of the relaxations used to solve the NP-hard optimization problems. In this paper, we pose batch selection as an NP-hard integer programming problem and propose two convex relaxations to solve the same. More importantly, we establish concrete mathematical guarantees on the quality of the relaxations. We now describe the mathematical formulations of the two algorithms.

3 The Proposed Batch Mode Active Learning Formulation

Consider a batch mode active learning problem, where we are given a training set L_t and an unlabeled set U_t at time t. Let w^t be the classifier trained on L_t. The objective is to select a batch B (containing k points) from U_t in such a way that the future learner w^{t+1}, trained on $L_t \bigcup B$, has maximum generalization capability. Let Y denote the set of possible classes in the problem. We quantify the quality of a batch of selected samples based on their informativeness and diversity, that is, we would like to select a batch of samples such that each point furnishes valuable information and the selected samples have minimal redundancy among them.

Formally, we compute an information vector $c \in \Re^{|U_t| \times 1}$ where $c(i)$ denotes the information furnished by the point x_i in the unlabeled pool. The uncertainty of the trained model w^t on a point x_i is used as a measure of the informativeness of x_i; higher uncertainty values denote higher degrees of information. The uncertainty of an unlabeled point x_i (that is, $c(i)$) is computed as the entropy $S(y|x_i, w^t)$ of the distribution $P(y|x_i, w^t)$, such that

$$c(i) = S(y|x_i, w^t) = -\sum_{y \in Y} P(y|x_i, w^t) \log P(y|x_i, w^t). \tag{1}$$

In addition to c, a divergence matrix $R \in \Re^{|U_t| \times |U_t|}$ is also defined whose (i, j)th entry is a measure of redundancy between unlabeled points x_i and x_j (higher the value of R_{ij}, lower the redundancy). The divergence measure between two points is an estimate of the amount of information overlap between the points, which is captured by the symmetric Kullback Leibler divergence. Let p^i and p^j denote the vectors of posterior probabilities of two points x_i and x_j in the unlabeled pool with respect to all the classes. Then, the (i, j)th entry in matrix R is equal to the symmetric KL divergence between the two vectors of probability values [29]:

$$R(i, j) = \sum_{y=1}^{|Y|} \left(p_y^i - p_y^j\right) \log \frac{p_y^i}{p_y^j}. \tag{2}$$

By definition, all the entries in c and R are non-negative, that is, $c_i \geq 0$ and $r_{ij} \geq 0$. Also, $R_{ii} = 0, \forall i$. Given c and R, our objective is to select a batch of points having high information scores and high divergence (or minimal redundancy) among them. For notational simplicity, we combine c and R into a single matrix $D \in \Re^{|U_t| \times |U_t|}$ as follows:

$$D(i, j) = \begin{cases} R(i, j), & \text{if } i \neq j, \\ \lambda c(i), & \text{if } i = j, \end{cases} \tag{3}$$

where each entry in the matrix D is non-negative, that is $d_{ij} \geq 0, \forall i, j$. λ is a trade-off parameter. We note that the matrix D can be defined suitably based on the application at hand. Without any loss of generality, we proceed with the criterion based on entropy and redundancy and explain our framework.

We now formulate the batch selection task as an explicit mathematical optimization problem, where the objective is to select a batch of points with high aggregate uncertainty scores and high divergences among them. Specifically, we define a binary vector m with $|U_t|$ entries ($m \in \{0, 1\}^{|U_t| \times 1}$) where each entry m_i denotes whether the corresponding unlabeled point x_i will be included in the batch ($m_i = 1$) or not ($m_i = 0$). Our batch selection criterion (with given batch size k) can thus be expressed using the following integer quadratic programming problem:

$$\max_m m^T D m$$
$$\text{s.t.} \quad m_i \in \{0, 1\}, \forall i \quad \text{and} \quad \sum_{i=1}^{|U_t|} m_i = k. \tag{4}$$

The binary constraint on m_i makes this IQP problem NP-hard. We note that a similar batch selection criterion, based on uncertainty and redundancy, has been used in previous research on multi-label learning [30]. However, the optimization problem was directly solved using standard quadratic programming solvers, after relaxing the integer constraints on m. Further, no theoretical guarantees were established on the quality of the obtained solution. Our primary contribution in this work is the derivation of two efficient convex relaxations to solve the above NP-hard IQP. We also establish concrete mathematical bounds on the quality of the solutions obtained using the convex relaxations. (A comparison of the proposed techniques against the QP-based solution strategy is included in the Supplemental File, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2015.2389848). We now discuss the two relaxations to solve this NP-hard problem.

4 BatchRank : Convex Relaxation I

We first show that the IQP in Equation (4) is equivalent to an integer linear programming (ILP) problem in Lemma 1.

Lemma 1. *The integer quadratic programming batch mode active learning formulation in Equation (4) can be simplified into an equivalent integer linear programming problem.*

Proof. We introduce a binary matrix $Z = (z_{ij})$ with $z_{ij} = m_i.m_j$. Thus, the optimization problem in (4) reduces to:

$$\max_{m,Z} \sum_{i,j} d_{ij} z_{ij}$$
$$\text{s.t.} \quad z_{ij} = m_i m_j, \quad \sum_{i=1}^{|U_t|} m_i = k, \quad \textit{and} \quad m_i \in \{0, 1\}, \forall i. \tag{5}$$

The quadratic equality constraint $z_{ij} = m_i m_j$ makes this problem difficult to solve. Interestingly, we can show that this quadratic constraint, in fact, allows itself to be represented as a simpler linear inequality $-m_i - m_j + 2z_{ij} \leq 0, \forall i, j$. This ensures that the value of z_{ij} is 0 if either m_i or m_j (or both) is equal to 0. When both m_i and m_j are equal to 1, z_{ij} is free to be either 0 or 1. However, the maximization criterion in (5) forces the value of z_{ij} to be 1 since $d_{ij} \geq 0$. Hence, the problem can now be written as:

$$\max_{m,Z} \sum_{i,j} d_{ij} z_{ij}$$
$$\text{s.t.} \quad -m_i - m_j + 2z_{ij} \leq 0, \forall i,j$$
$$\textit{and} \quad \sum_{i=1}^{|U_t|} m_i = k, m_i, z_{ij} \in \{0,1\}, \forall i,j. \tag{6}$$

This is an integer LP problem, proving Lemma 1. □

Although a global maximum exists for the ILP, it is computationally expensive to compute. To solve such an ILP, a standard approach is to employ the LP relaxation.

Lemma 2. *The convex LP relaxation of the above ILP (Equation (6)) in Lemma* 1 *is equivalent to a ranking formulation based on the entries in the matrix D.*

Proof. We consider the following linear program relaxation:

$$\max_{m,Z} \sum_{i,j} d_{ij} z_{ij}$$
$$\text{s.t.} \quad -m_i - m_j + 2z_{ij} \leq 0, \forall i,j, \quad \sum_{i=1}^{|U_t|} m_i = k$$
$$\textit{and} \quad m_i, z_{ij} \in [0,1], \forall i,j. \tag{7}$$

Since this is a maximization problem with $d_{ij} \geq 0$, at optimality, $z_{ij} = \frac{m_i + m_j}{2}$ (from the inequality constraint $-m_i - m_j + 2z_{ij} \leq 0$). Hence, (7) is equivalent to:

$$\max_{m} \frac{1}{2} \sum_{i,j} d_{ij}(m_i + m_j)$$
$$\text{s.t.} \quad \sum_{i=1}^{|U_t|} m_i = k \quad \textit{and} \quad m_i \in [0,1], \forall i. \tag{8}$$

This formulation admits an analytical (as well as an integer) solution for m by a simple ranking based on the entries in the matrix D. The objective in (8) can be written as $\sum_{i,j} d_{ij} m_i + \sum_{i,j} d_{ij} m_j$. Since the matrix D is symmetric, the maximization problem essentially becomes equivalent to ranking the column sums of D (hence the name BatchRank). This proves Lemma 2. □

4.1 Solution Bound of BatchRank

In this section, we prove a bound on the solution to the convex LP relaxation in (8) with respect to the solution of the original NP-hard integer quadratic programming problem. To this end, we transform the original maximization problem in Equation (4) into an equivalent minimization problem through the following objective function:

$$f(m) = ||D||_1 - m^T D m, \tag{9}$$

where $||D||_1 = \sum_{i,j} d_{ij}$. We note that since $||D||_1$ is constant for a given matrix D, maximizing $m^T D m$ as in Equation (4) is equivalent to minimizing the function $f(.)$ defined above, that is, maximizing $m^T D m$ and minimizing $f(.)$ as defined above will fetch the same solution to the variable m. Since we are interested in the solution quality of m, we prove an upper bound on the minimization problem in Equation (9). Since the solution to m is the same, it is essentially equivalent to proving a bound on the original maximization problem in Equation (4). The main result is summarized in the following theorem:

Theorem 1. *Let m^* and $\widehat{m}$ be the optimal solutions of the original NP-hard IQP in Equation (4) and the convex relaxation in Equation (8), respectively. Then,*

$$f(\widehat{m}) \leq 2f(m^*).$$

Proof. The optimization in (8) is an LP relaxation of the quadratic formulation in (4) and thus the objective value of (8) is larger than that of (4). That is,

$$m^{*T} D m^* \leq \frac{1}{2} \sum_{i,j} d_{ij}(\widehat{m_i} + \widehat{m_j})$$
$$= \frac{1}{2} \sum_{i,j:\widehat{m_i}+\widehat{m_j}=1} d_{ij} + \sum_{i,j:\widehat{m_i}+\widehat{m_j}=2} d_{ij}. \tag{10}$$

Since all entries in D are non-negative, the following holds:

$$\begin{aligned} ||D||_1 &= \sum_{ij} d_{ij} \\ &= \sum_{i,j:\widehat{m_i}+\widehat{m_j}=2} d_{ij} + \sum_{i,j:\widehat{m_i}+\widehat{m_j}=1} d_{ij} + \sum_{i,j:\widehat{m_i}+\widehat{m_j}=0} d_{ij} \\ &\geq \sum_{i,j:\widehat{m_i}+\widehat{m_j}=2} d_{ij} + \sum_{i,j:\widehat{m_i}+\widehat{m_j}=1} d_{ij}. \end{aligned}$$

Thus,

$$\sum_{i,j:\widehat{m_i}+\widehat{m_j}=1} d_{ij} \leq ||D||_1 - \sum_{i,j:\widehat{m_i}+\widehat{m_j}=2} d_{ij}. \tag{11}$$

Combining the above two, we have

$$\begin{aligned} f(m^*) &= ||D||_1 - m^{*T} D m^* \\ &\geq ||D||_1 - \frac{1}{2} \sum_{i,j:\widehat{m_i}+\widehat{m_j}=1} d_{ij} - \sum_{i,j:\widehat{m_i}+\widehat{m_j}=2} d_{ij} \\ &\geq ||D||_1 - \frac{1}{2}\left(||D||_1 - \sum_{i,j:\widehat{m_i}+\widehat{m_j}=2} d_{ij}\right) - \sum_{i,j:\widehat{m_i}+\widehat{m_j}=2} d_{ij} \\ &= \frac{1}{2}\left(||D||_1 - \sum_{i,j:\widehat{m_i}+\widehat{m_j}=2} d_{ij}\right) = \frac{1}{2} f(\widehat{m}). \end{aligned}$$

The first inequality follows from Equation (10) and the second from Equation (11). The last equality is true because in the evaluation of $m^T D m$, only the indices where both m_i and m_j are 1 will survive, others will vanish. This completes the proof of the theorem. We thus note that the convex relaxation of the original NP-hard problem in BatchRank has a guaranteed bound on the solution quality. □

4.2 The Iterative Truncated Power Algorithm

As evident from Lemma 2, the LP relaxation of the NP-hard IQP in Equation (4) reduces to selecting a set of k unlabeled points producing the k largest column sums of the matrix D. To further improve the solution, we use the iterative truncated-power algorithm proposed by Yuan and Zhang [31]. This solution was proposed in the context of the sparse eigenvalue problem and was also shown to be applicable to the densest k-subgraph problem (DkS). Mathematically, DkS can be expressed as a binary quadratic programming problem, equivalent to Equation (4). Starting with an initial approximation x_0, the algorithm generates a sequence of solutions $x_1, x_2, \ldots$. At each time step t, the vector x_{t-1} is multiplied by the weight matrix D and then the entries are truncated to zeros except for the k largest entries, which becomes the new solution x_t. This process is repeated until convergence. This simple, yet efficient algorithm has a guaranteed monotonic convergence for a positive semi-definite (psd) weight matrix D. When the matrix D is not psd, the algorithm can be run on the shifted quadratic function (with a positive scalar added to the diagonal elements) to guarantee a monotonic convergence [31].

We use the BatchRank solution as the initial approximation x_0 followed by the iterative truncated power method to derive the final set of unlabeled samples to be selected for manual annotation. Since the convergence is monotonic, the quality of the solution can only improve over the iterations and the bound established in Theorem 1 on the quality of the solution still holds. Moreover, the running time increases only marginally due to the iterative process as it involves minimal computational overhead and converges fast. The pseudo-code for the BatchRank algorithm is given in Algorithm 1. The complexity of the algorithm is $O(n^2)$, where n is the number of unlabeled samples.

Algorithm 1. BatchRank Algorithm for Batch Mode Active Learning

```
Require: Training set $L_t$, Unlabeled set $U_t$ and batch size $k$
Train a classifier $w^t$ on the training set $L_t$
Compute information vector $c$ (Equation (1)) and the divergence matrix $R$ (Equation (2)) using $w^t$
Compute the matrix $D$, as described in Equation (3)
Compute a vector $v \in \Re^{|U_t| \times 1}$ containing the column sums of $D$
Identify the $k$ largest entries in $v$ and derive the initial solution $x_0$
t = 1
repeat
    Compute $x_t^{'} = D.x_{t-1}$
    Identify $F_t$ as the index set of $x_t^{'}$ with top $k$ values
    Set $x_t$ to be 1 on the index set $F_t$ and 0 otherwise
    t = t + 1
until Convergence
Select a batch of $k$ unlabeled samples based on the final solution $x_t$
```

5 BatchRand: Convex Relaxation II

5.1 Problem Formulation

In this section, we attempt to further improve the theoretical guarantee of the solution quality of the NP hard batch selection problem. We therefore propose a second relaxation based on a randomized approximation algorithm (hence the name BatchRand). Starting with Equation (4), we first make the following variable transformation:

$$y_i = 2\left(m_i - \frac{1}{2}\right) \Rightarrow m_i = \frac{y_i + 1}{2}$$
$$\Rightarrow \sum_{i=1}^{n} y_i = 2\sum_{i=1}^{n} m_i - n = 2k - n \triangleq p,$$

where $n = |U_t|$ is the number of unlabeled data samples in the unlabeled pool. The entire optimization problem in Equation (4) is now rewritten in terms of the new variable y (ignoring the constant $\frac{1}{4}$):

$$\max_{y} \sum_{i,j} d_{ij}(y_i + 1)(y_j + 1)$$
$$\text{s.t.} \quad y_i \in \{-1, 1\}, \forall i \quad \text{and} \quad \sum_{i=1}^{|U_t|} y_i = p. \tag{12}$$

5.2 Multi-Dimensional Relaxation

Since solving the integer quadratic program in Equation (12) is NP hard, we consider relaxations of the constraints. Specifically, we follow the strategy proposed by Goemans and Williamson [32], where each variable y_i is relaxed to a multi-dimensional vector v_i belonging to $\Re^n$ of unit euclidean norm, instead of a one dimensional scalar variable. In other words, we assume that each vector v_i belongs to the n-dimensional unit sphere S_n. The relaxation of the NP hard problem in Equation (12) is therefore given by (ignoring the constant 1):

$$\max_{v} \sum_{i,j} d_{ij}\left(v_i^T v_j + v_0^T v_i + v_0^T v_j\right) \tag{13}$$

$$\text{s.t.} \quad v_i \in S_n, \forall i \quad \text{and} \quad \sum_{i=1}^{|U_t|} v_i = p, \tag{14}$$

where v_0 is a vector of n-dimensions with all entries 1. (The equality constraint in Equation (14) implies that the sum across all the entries in all the v_i vectors should equal the scalar constant p. This is because each scalar variable y_i was relaxed into a vector variable v_i and the sum of all y_i was constrained to be equal to p). Once we solve for the vectors v from this formulation (we present the solution details below), we select a random unit vector r that is uniformly distributed on the unit sphere and find the dot product of r with every vector v_i. We then select the set of unlabeled points whose corresponding v vectors yield a positive dot product with the random unit vector r, as in [32]. However, we note that the number of positive dot products may not exactly equal k, which means that the number of instances selected by BatchRand may not equal the pre-specified batch size.

5.3 Semi-Definite Programming Relaxation

Using the decomposition $Q = B^T B$, we note that any positive semi-definite matrix with diagonal entries 1 corresponds to a set of unit vectors v_i if we correspond the vector v_i to the ith column of the matrix B. Then, $q_{ij} = v_i v_j$, which accounts for the term $v_i^T v_j$ in the objective function of Equation (13). However, to incorporate the terms $v_0^T v_i$ and $v_0^T v_j$ in the objective, the matrix Q is decomposed as

$$Q = \begin{pmatrix} v_0^T \\ B^T \end{pmatrix} \begin{pmatrix} v_0 & B \end{pmatrix},$$

where $B = [v_1 \, v_2 \, \ldots v_n]$. We can therefore rewrite the entire relaxation in terms of the defined matrix Q (in the previous equation) as follows:

$$\begin{aligned} \max_Q \sum_{i,j} & d_{i,j}(Q_{i+1,j+1} + Q_{1,i+1} + Q_{1,j+1}) \qquad (15) \\ \text{s.t. } & Q_{ii} = 1, \text{for i} = 2 \text{ to n} + 1, \\ & \sum_{j=2}^{n+1} Q_{1j} = p, \; Q_{11} = n \\ & \text{and } Q \succeq 0 \;\; (\text{Q is psd}). \end{aligned}$$

This is a semi-definite programming problem and can be solved using existing software packages like SeDuMi. The pseudocode of the BatchRand algorithm is provided below. The complexity of the algorithm is $O(n^3)$, n being the number of unlabeled data samples.

Algorithm 2. BatchRand Algorithm for Batch Mode Active Learning

```
Require: Training set $L_t$, Unlabeled set $U_t$ and batch size $k$
Train a classifier $w^t$ on the training set $L_t$
Compute information vector $c$ (Equation (1)) and the divergence matrix $R$ (Equation (2)) using $w^t$
Compute the matrix $D$, as described in Equation (3)
Solve the SDP relaxation problem (Equation (15)) to yield a set of vectors $v$
Select a random unit vector $r$ and evaluate the dot product of $r$ with each vector $v_i$
Select the set of unlabeled points $S = \{i | v_i.r \geq 0\}$ for manual annotation
```

5.4 Probabilistic Solution Guarantee

We first rewrite the objective in Equation (13) in a simplified form as follows:

$$\sum_{i,j} d_{ij}\left(v_i^T v_j + v_0^T v_i + v_0^T v_j\right) = \sum_{i,j} \widehat{d_{ij}}\left(v_i^T v_j\right),$$

where v_0 is the vector obtained from the first row of the decomposed matrix after solving the SDP problem and $\widehat{d_{ij}}$ is obtained from d_{ij}, to simplify the representation. The main result regarding the solution bound of the BatchRand algorithm is summarized in the following theorem:

Theorem 2. *Let W denote the value of the objective function produced using the BatchRand algorithm and $E(W)$ denote its expectation. Also, let $\widehat{D}_{total}$ denote the sum of all entries in the matrix $\widehat{D}$. Then,the following bound holds:*

$$[E(W) + \widehat{D}_{total}] \geq 0.87856 \left[\sum_{i,j} \widehat{d_{ij}} v_i v_j + \widehat{D}_{total} \right].$$

Proof. Consider a random unit vector r. By the linearity of expectation, the expectation of the value of the objective function is given by (we drop the sub-scripts i and j from the summation for notational convenience):

$$\begin{aligned} E(W) &= \sum \widehat{d}_{ij}[1.Pr(sgn(v_i.r) = sgn(v_j.r)) \\ &\quad + (-1).Pr(sgn(v_i.r) \neq sgn(v_j.r))] \\ &= \sum \widehat{d}_{ij}[1 - 2.Pr(sgn(v_i.r) \neq sgn(v_j.r))] \\ &= \sum \widehat{d}_{ij}\left[1 - 2\frac{arccos(v_i.v_j)}{\pi}\right], \end{aligned}$$

where $sgn(x) = 1$ if $x \geq 0$ and -1 otherwise. The last equality follows from the result proved by Goemans and Williamson [32]. Further, it can be shown that for $-1 \leq z \leq 1, 1 - \frac{arccos(z)}{\pi} \geq \alpha.\frac{1}{2}(1+z)$, where

$$\alpha = \min_{0 \leq \theta \leq \pi} \frac{2}{\pi}.\frac{\theta}{1 - \cos\theta} \geq 0.87856,$$

(the proof of the above inequality can be found in [32]). That is, $1 - 2\frac{arccos(z)}{\pi} \geq \alpha(1+z) - 1$. Since in our formulation, the $v_i s$ are all unit vectors, we have $-1 \leq z = v_i v_j \leq 1, \;\; \forall i, j.$ Therefore, $1 - 2\frac{arccos(v_i v_j)}{\pi} \geq \alpha(1 + v_i v_j) - 1$.

Combining all of the above, we get

$$\begin{aligned} E(W) &\geq \sum \widehat{d}_{ij}[\alpha(1 + v_i v_j) - 1] \\ &= \alpha \sum \widehat{d}_{ij} v_i v_j + (\alpha - 1)\widehat{D}_{total} \;\Rightarrow [E(W) + \widehat{D}_{total}] \\ &\geq \alpha \left[\sum_{i,j} \widehat{d}_{ij} v_i v_j + \widehat{D}_{total} \right], \end{aligned}$$

which proves the theorem. □

These bounds corroborate the fact that the convex relaxations that we use to solve the NP-hard batch selection problem actually produce high quality solutions—that is, the solutions we get through the convex relaxations are very close to the best possible (optimal) solution. From an active learning point of view, this implies that the set of unlabeled samples we select for manual annotation using the convex relaxations very closely resembles the best possible set of samples for a given batch selection criterion.

5.5 Computational Considerations

We note that the time complexity of BatchRank is $O(n^2)$ and that of BatchRand is $O(n^3)$. This may limit the scalability of the algorithms to very large datasets. To overcome this, we used a sub-sampling strategy in our empirical study, where the current classifier was applied to all the unlabeled data samples and the batch selection was restricted to the top p uncertain samples (given by the entropy values of the current model). The value of p was taken as 400 in our experiments and can be suitably selected based on a given application. In future, we plan to investigate other sub-sampling strategies; for instance, another sub-sampling approach may be to perform a k-means clustering on the unlabeled data with $k = p$, take the p cluster centers as the sub-sampled pool and restrict the batch selection to this subset. This method will select the representative samples from the unlabeled set into the subsampled pool, as opposed to selecting the uncertain samples, as in our experiments. Comparing the two approaches is an interesting direction of future research.

6 Experiments and Results

6.1 Datasets and Feature Extraction

In this section, we present the details of the datasets and the feature extraction process used in our experiments.

UCI datasets. We used nine datasets (binary and multiclass) from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/) as benchmarks to validate our algorithms.

Face recognition datasets. We also used two real-world biometric datasets in our experiments: (1) the VidTIMIT dataset [33], which contains video recordings of subjects reciting short sentences under unconstrained natural conditions and (2) the MOBIO dataset [34], which was recently created for the Mobile Biometry (MOBIO) challenge to test state-of-the-art face and speech recognition algorithms. It contains recordings of subjects under challenging real world conditions, captured using a hand-held device. (Our purpose was to test the performance of active learning and so, for the MOBIO dataset, we did not follow the protocols specified in the actual challenge, which were intended for person recognition.) Both these datasets contain video recordings of subjects under natural conditions where there is a redundancy of information and are therefore appropriate to test active learning algorithms. The face images in the video frames were automatically detected and cropped to 128 by 128. The discrete cosine transform feature was extracted from the face images [35] and PCA was used to reduce the dimension to 100, retaining about 99 percent of the variance.

Facial expression recognition datasets. We further used two challenging facial expression recognition datasets—the MMI and the MindReading datasets, to test our algorithms. The MMI dataset contains videos of subjects exhibiting various expressions and is extensively used in expression recognition research [36]. The MindReading is a computer based guide to emotions primarily collected to help individuals diagnosed with autism recognize facial expressions of emotion [37]. Both these datasets contain videos of subjects under challenging real world conditions and thereby represent an appropriate ground to test our algorithms for active learning in facial expression recognition. Videos containing the six basic expressions for 30 subjects were selected from the databases. Relevant frames around the peak of the expression were extracted from each video. Automated facial detection [38] was applied to crop the faces. The Gabor filter was applied to the images for feature extraction ([39]) and PCA was used to reduce the dimensionality to 100, retaining about 98 percent of the variance.

Multi-label datasets. In addition, we also validated our algorithms on two multi-label datasets—the Scene and the Yeast. These are widely used in multi-label learning research [40].

6.2 Competing Algorithms and Experiment Set-Up

We compared the proposed algorithms against the following batch mode active learning strategies proposed in the literature: (1) *Random*, where a batch of points is queried at random from the unlabeled set (this is used for baseline comparison), (2) *Most Uncertain*, where the top k uncertain points are queried from the unlabeled set, k being the batch size, (3) *Fisher*, that selects a batch of samples for manual annotation by maximizing the Fisher information of the classification model (proposed by Hoi et al. [1]), (4) *Disc*, a discriminative strategy that selects a batch of points by optimizing the performance of the future learner, proposed by Guo and Schuurmans [4] and (5) *Matrix*, that queries a batch of unlabeled points by maximizing the mutual information between the labeled and unlabeled points [5]. The Disc and Matrix approaches have been shown to be the state-of-the-art BMAL schemes [5]. We refer to BatchRank followed by the Truncated Power iterative method as simply BatchRank in our experiments.

For each dataset, we started with an initial labeled training set, an unlabeled pool and a test set. For a fixed batch size k, each algorithm repeatedly selected k instances from the unlabeled pool to be labeled each time (as mentioned in Section 5.2, the number of points queried by BatchRand may not be exactly k, this algorithm was therefore used first to note the exact number of samples queried in each iteration; these values were then used in the other algorithms to query the same number of points in the corresponding iteration, for fair comparison). After each batch selection, the selected points were removed from the unlabeled pool and appended to the training set. The goal was to study the improvement in performance on the test set with an increasing size of the training set (this experimental setup is similar to earlier work [4], [5]). All the results were averaged over 10 runs to rule out the effects of randomness. The sub-sampling strategy, mentioned in Section 5.5, was used for the BatchRand and BatchRank algorithms when the size of the unlabeled set was more than 400. Logistic regression was used as the base classifier (similar to [4]). The training, unlabeled and test splits for each dataset are summarized in Table 1. The algorithms were implemented in MATLAB on a quad-core Intel processor with 2.66 GHz CPU and 8 GB RAM. The entries in the uncertainty vector c and the redundancy matrix D were normalized between 0 and 1. The weight parameter λ was selected as 1 based on preliminary empirical studies. The batch size k was fixed at 10 in our experiments, for all the datasets except the Wine, Spect and Heart datasets, where it was taken as 5, due to the small sizes of these datasets.

6.3 Batch Mode Active Learning on Binary and Multi-Class Datasets

The results on the UCI datasets are reported in Fig. 1. In each graph, the x-axis denotes the size of the labeled training set and the y-axis denotes the accuracy obtained on the test set. As mentioned earlier, the objective was to study the growth in accuracy on the test set as more and more points are queried from the unlabeled set.

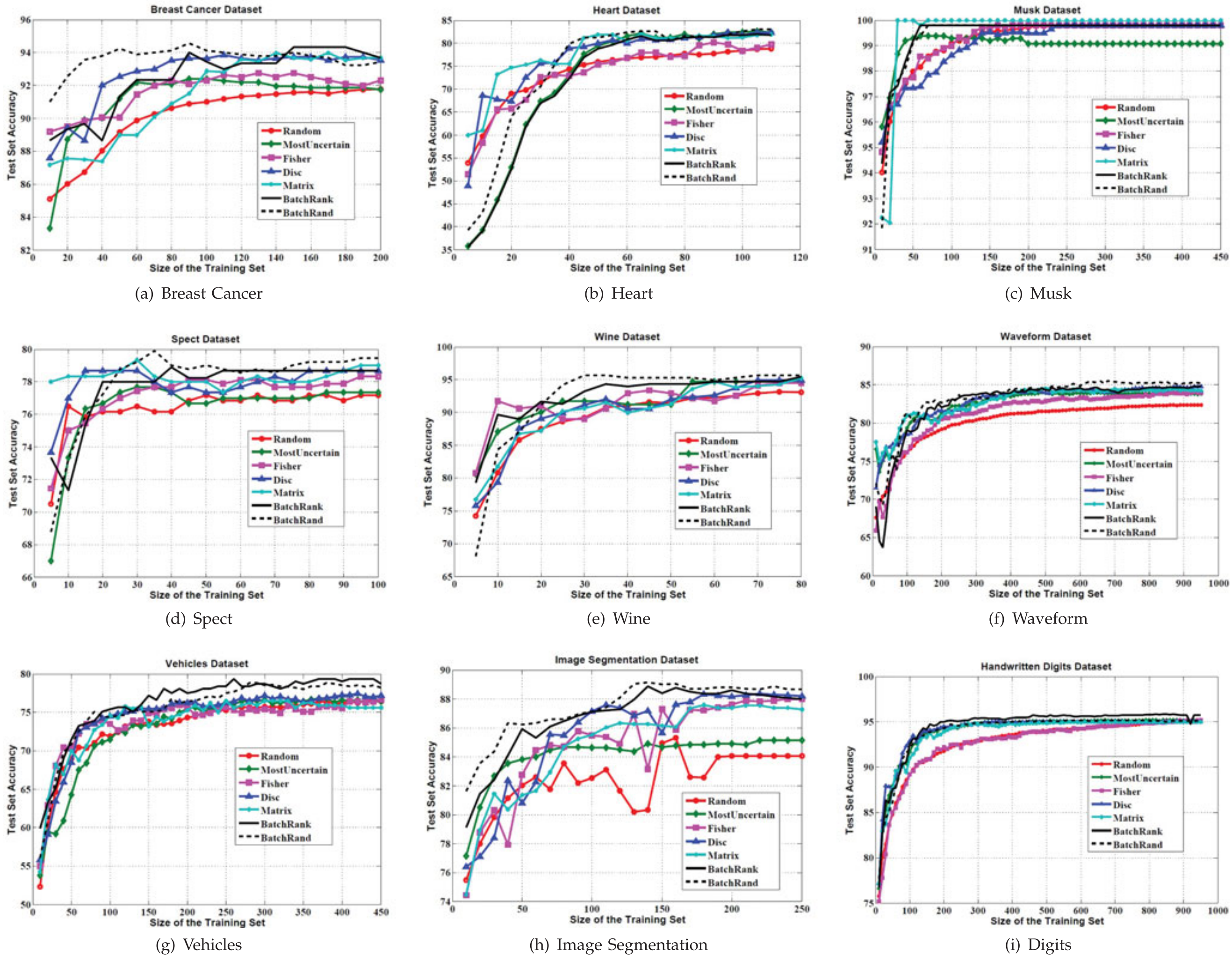
From the results, it is evident that BatchRand and BatchRank outperform Random sampling on all the datasets, as the accuracy grows at a faster rate with increasing size of the labeled set. This shows that the proposed approaches succeed in selecting the salient and prototypical samples

TABLE 1
Dataset Details

Dataset	Classes	Dimensionality	Training Set Size	Unlabeled Set Size	Testing Set Size
Breast Cancer	2	30	10	259	300
Heart	2	13	4	120	146
Musk	2	166	2	500	490
Spect	2	22	7	110	150
Wine	3	13	3	87	88
Waveform	3	20	9	1,000	500
Vehicles	4	18	16	500	330
Image Segmentation	7	19	35	300	2,000
Handwritten Digits	10	64	50	1,000	2,751
VidTIMIT	25	100	250	1,000	4,500
MOBIO	25	100	50	1,000	4,500
MindReading	6	100	50	1,000	1,511
MMI	6	100	50	1,000	1,785
Scene	6	294	10	350	2,047
Yeast	14	103	10	600	1,807

from the unlabeled data population and attain a given level of accuracy with the least number of labeled examples. The Most Uncertain and Fisher methods perform better than random sampling, but are not as good as the proposed algorithms. The proposed frameworks consistently depict comparable performance to Disc and Matrix, the state-of-the-art BMAL techniques.

Fig. 2 depicts the results on the face recognition and facial expression recognition datasets. We note that our algorithms once again depict comparable performance as

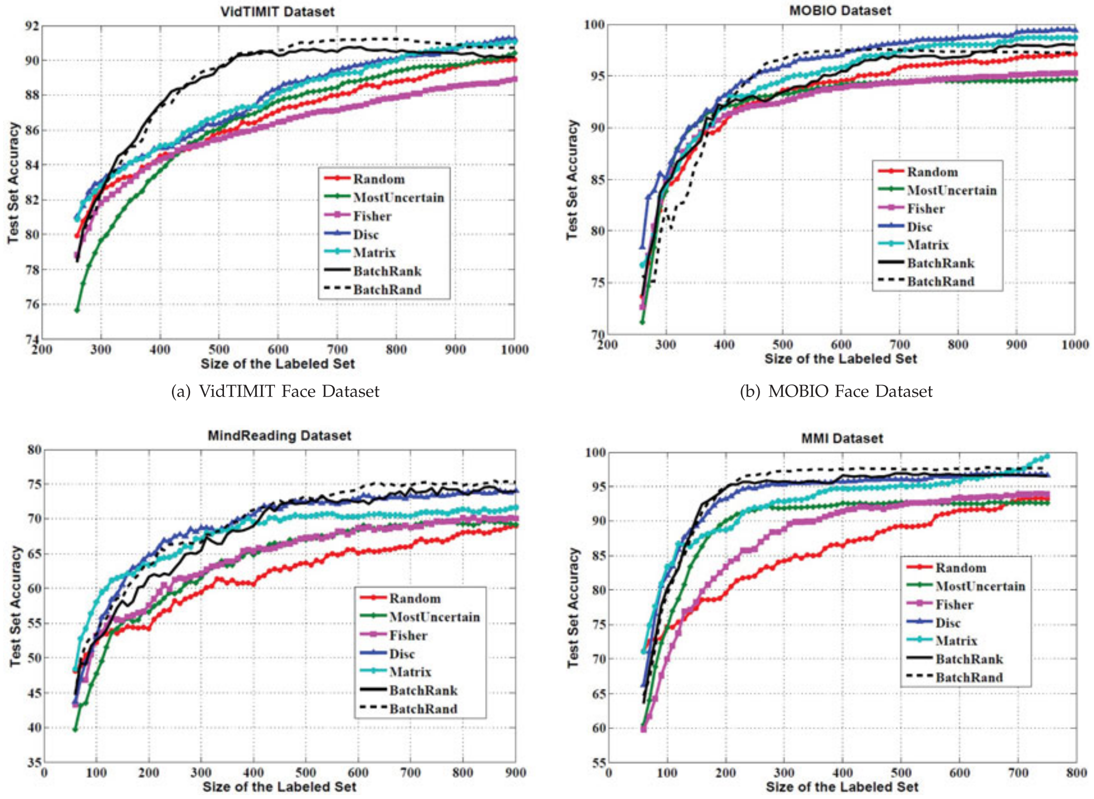


(a) Breast Cancer (b) Heart (c) Musk

(d) Spect (e) Wine (f) Waveform

(g) Vehicles (h) Image Segmentation (i) Digits

Fig. 1. Batch mode active learning on the UCI datasets. (Figures best viewed in color.)



(a) VidTIMIT Face Dataset

(b) MOBIO Face Dataset

(c) MindReading Expression Dataset

(d) MMI Expression Dataset

Fig. 2. Batch mode active learning face recognition and facial expression recognition datasets. (Figures best viewed in color.)

the state-of-the-art (the BatchRand method in fact, marginally outweighs Disc and Matrix on the VidTIMIT and MindReading datasets). Moreover, our methods are computationally much more efficient than the state-of-the-art (depicted in Section 6.5). We also note that the random sampling method may sometimes depict good performance, as in the VidTIMIT and MOBIO datasets. However, it is not consistent and performs poorly in the other datasets.

6.4 Batch Mode Active Learning on Multi-Label Datasets

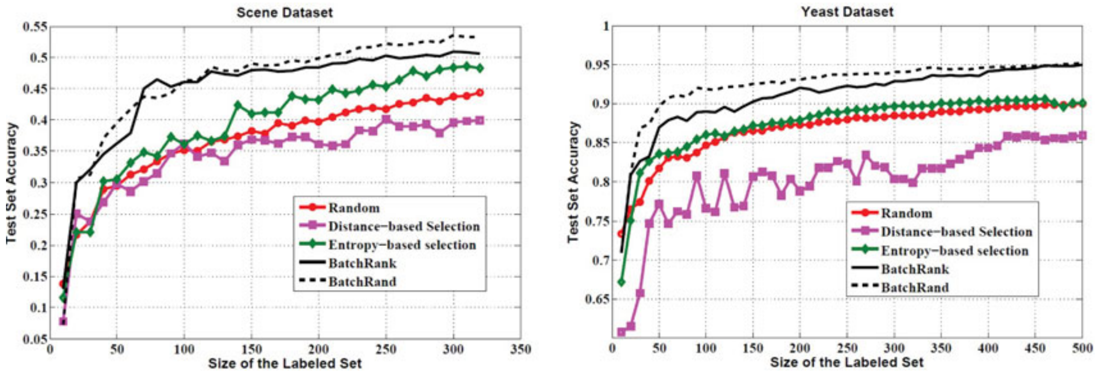
Multi-label learning is a generalization of conventional single-label learning, where each data sample can have multiple labels associated with it. Manual annotation of samples is even more difficult in a multi-label application, as the user has to scan through all the possible labels to decide the label set of a particular example. Thus, batch mode active learning is of paramount importance in such settings. The proposed BatchRank and BatchRand frameworks are flexible and can be extended to multi-label contexts. We demonstrate their performance on two benchmark multi-label datasets in this section. For this purpose, the entropy term in the matrix D was modified and was computed as the average entropy over the individual classes:

$$S(Y|x_i, w^t) = \frac{1}{|Y|}\sum_{j=1}^{|Y|}\left[p_j^i \log p_j^i + \left(1 - p_j^i\right)\log\left(1 - p_j^i\right)\right].$$

The proposed approaches were compared against (i) *Random Selection*, (ii) *Distance based Selection*, in which an SVM was trained for each possible class and the k closest unlabeled points (k being the batch size) to all the hyperplanes in the feature space were selected for annotation, (as in [41]) and (iii) *Entropy based Selection*, where the entropy was computed for every unlabeled instance and the top k points were queried based on the entropy ranking (as in [42]). A polynomial kernel SVM was used as the base classifier because of its established performance in multi-label learning [41]. The results on two multi-label datasets, Scene (with 6 classes) and Yeast (with 14 classes) are shown in Fig. 3. The results corroborate the conclusions drawn in the previous experiments.

6.5 Run Time Analysis

In this section, we perform an analysis of the computation time of each of the batch mode active learning algorithms. Table 2 (binary and multi-class datasets) and Table 3 (multi-label datasets) report the average time taken to query a batch of samples from an unlabeled set for all the algorithms. We note that random selection and uncertainty based selection are the most efficient in terms of running time. The Fisher information based selection framework also has low computation time. The proposed algorithms BatchRank and BatchRand surpass Disc by a substantial margin in terms of running time. The improvement is more prominent in case of larger datasets (VidTIMIT, MOBIO and Handwritten Digits) thus demonstrating their potential for large scale learning. The Disc algorithm involves intensive computation and classifier retraining as part of the optimization process, which adversely affects its running time. The Matrix algorithm is better than Disc (as observed in [5]) but is slower as compared to BatchRank and BatchRand.



(a) Multi-label Scene dataset

(b) Multi-label Yeast dataset

Fig. 3. Multi-label batch mode active learning on the scene and yeast datasets. (Figures best viewed in color.)

TABLE 2
Time Taken (in Seconds) to Query a Batch of Samples from an Unlabeled Set

Dataset	Unlabeled Set	Random	MostUncertain	Fisher	Disc	Matrix	BatchRank	BatchRand
Wine	87	0.01	0.06	0.49	6.92	1.02	0.25	0.86
Spect	110	0.01	0.11	0.78	1.18	2.34	0.31	3.91
Heart	120	0.01	0.08	0.33	5.21	1.80	0.30	0.37
Breast Cancer	259	0.01	0.23	0.46	16.23	5.78	0.93	0.49
Image Segmentation	300	0.01	0.25	0.64	12.09	3.61	1.99	1.44
Musk	500	0.01	0.12	1.03	96.14	35.51	3.24	2.17
Vehicles	500	0.01	0.18	0.97	27.04	9.87	3.79	2.94
Waveform	1,000	0.01	0.58	1.03	296.43	122.34	9.28	11.38
Handwritten Digits	1,000	0.01	0.53	1.56	977.28	234.78	15.48	19.47
VidTIMIT	1,000	0.01	1.78	3.92	923.65	171.88	30.46	26.08
MOBIO	1,000	0.01	1.18	2.37	757.43	204.57	19.63	24.69
MindReading	1,000	0.01	1.48	1.78	88.71	12.23	12.71	13.32
MMI	1,000	0.01	0.44	1.86	73.94	129.77	12.09	11.94

Binary and multi-class datasets.

TABLE 3
Time Taken (in Seconds) to Query a Batch of Samples from an Unlabeled Set

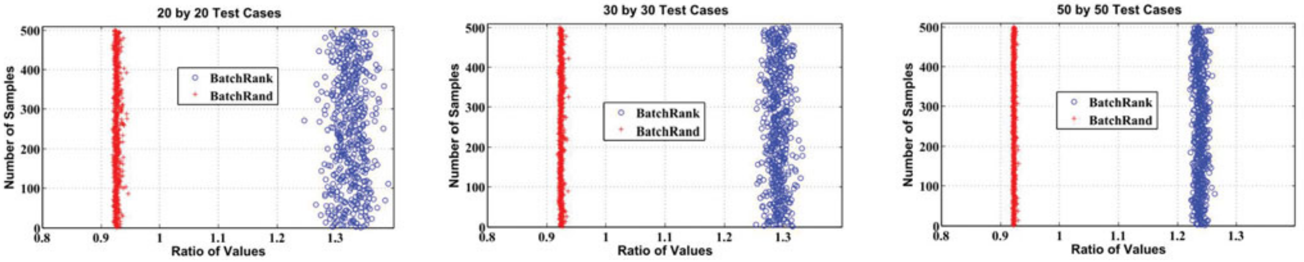
Dataset	Size of Unlabeled Set	Random	Brinker	Most Uncertain	BatchRank	BatchRand
Scene	350	0.01	0.72	1.39	2.91	2.28
Yeast	600	0.01	0.54	1.89	5.06	5.74

Multi-label datasets.

The results unanimously point to the conclusion that the proposed approaches deliver comparable performance as the state-of-the-art BMAL algorithms at a significantly lower computation time.

6.6 Validation of the Solution Bounds

To empirically validate the solution bounds of the proposed algorithms, we generated random symmetric matrices D, with $d_{ij} \geq 0$. We derived the optimal solutions m^* and $\widehat{m}$ by solving the integer programming problem in (4) and the relaxed formulation in (8) respectively for the BatchRank algorithm. Note that the IQP formulation in (4) can be solved exactly for small-scale problems. The ratio $\frac{f(\widehat{m})}{f(m^*)}$ was computed for the given matrix D and for a specific batch size k, where $f(.)$ was the function defined in Equation (9). We also computed the ratio $\frac{E(W)+\widehat{D}_{total}}{\sum \widehat{d_{ij}v_iv_j}+\widehat{D}_{total}}$ as described in Theorem 2 for the BatchRand algorithm. Fig. 4 shows sample results obtained on three test cases with different matrix dimensions (the batch size k was taken as 10 in all cases). Each graph shows the ratio $\frac{f(\widehat{m})}{f(m^*)}$ and $\frac{E(W)+\widehat{D}_{total}}{\sum \widehat{d_{ij}v_iv_j}+\widehat{D}_{total}}$ for 500 different matrices of the same dimension. For the BatchRank framework, it is evident that the ratio of the functional values is less than 2 in all cases, which validates the bound established in Section 4.1. Moreover, in all the test cases, the ratio is only slightly greater than 1 (in the range 1.2 to 1.4), which shows that the functional value obtained using the proposed method is very close to that obtained using the optimal solution. As mentioned in Section 4.1, this bound is on the quality of the initial approximation to the iterative process—further iterations will only improve the quality of the solution. In case of BatchRand, we note that the ratio is always greater than 87 percent validating the bound proved in Theorem 2. Further, we note that the ratio is actually very close to 93 percent in most cases depicting that the solutions obtained in practice are



(a) 20 × 20 Test Cases

(b) 30 × 30 Test Cases

(c) 50 × 50 Test Cases

Fig. 4. Validation of solution bounds of batchrank and batchrand. (Figures best viewed in color.)

much better than the theoretical guarantee. We therefore conclude that both BatchRank and BatchRand produce high quality solutions which very closely resemble the optimal solution to the original NP-hard IQP.

6.7 Noise Sensitivity

In many real-world scenarios, the labels of the data samples are noisy, either due to errors in data collection, or even because of human annotation errors. In this section, we study the label-noise sensitivity of the BatchRank and BatchRand algorithms. To simulate the situation, we artificially imparted stochastic labeling noise to the unlabeled samples. An n percent noise implies that the samples are randomly given an incorrect label with a probability of n percent. The labels of the test set were kept unchanged and the algorithms were run on clean as well as noisy data. We compared the proposed algorithms against random selection on the VidTIMIT dataset.

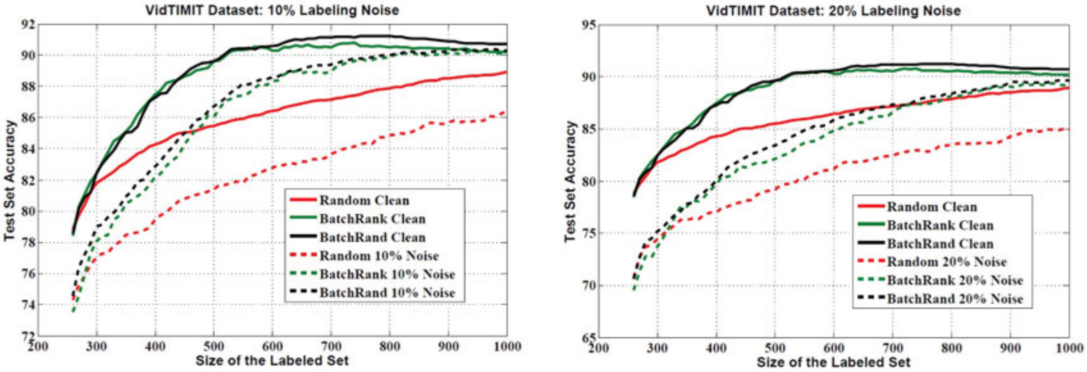
The results are depicted in Fig. 5, which plots the active learning curves for BatchRank, BatchRand and Random sampling with 10 and 20 percent labeling noise. As expected, the classification accuracy reduces in the presence of noise. But, from Fig. 5a, we note that, even with 10 percent labeling noise, the accuracy of both BatchRank and BatchRand drops only marginally as compared to the values on clean data and the final accuracy matches very closely to that obtained using clean data. Further, both the methods outperform random sampling on the same amount of noisy data and even on clean data. Even with 20 percent labeling noise, the final accuracy values of BatchRank and BatchRand are very close to those obtained using clean data. These results corroborate the fact that the proposed algorithms are robust to a significant amount of labeling noise.

A possible reason for this is the fact that both the BatchRank and BatchRand algorithms select unlabeled samples for query which are the most uncertain (and also diverse) with respect to the current classification model. Considering a binary classification problem, the uncertain samples typically lie close to the decision hyperplane. If the user gives an incorrect label of such an unlabeled sample, it will have minimal effect on the orientation of the decision boundary. In contrast, an incorrect label of a sample deep inside the region of a certain class can have much more severe effects on the decision boundary. The same argument can be extended to a multi-class classification problem with C classes by treating it as C binary classification problems using the one-versus-rest approach. Thus, the batch selection criterion of the proposed frameworks endow them with the capability to counter label noise and deliver high classification accuracy.

6.8 Class Imbalance

In real-world data, there is often a large variation in the number of samples belonging to the different classes. In this section, we study the performance of BatchRank and BatchRand to counter class imbalance. We once again present the results on the VidTIMIT dataset and compare the proposed algorithms against random selection. To simulate the real world scenario, an unlabeled pool of samples, with largely varying number of instances per class, was presented to the active learner for batch selection. The test set was kept unchanged. Fig. 6a depicts the percentage of images of each of the 25 classes that were present in the unlabeled pool.

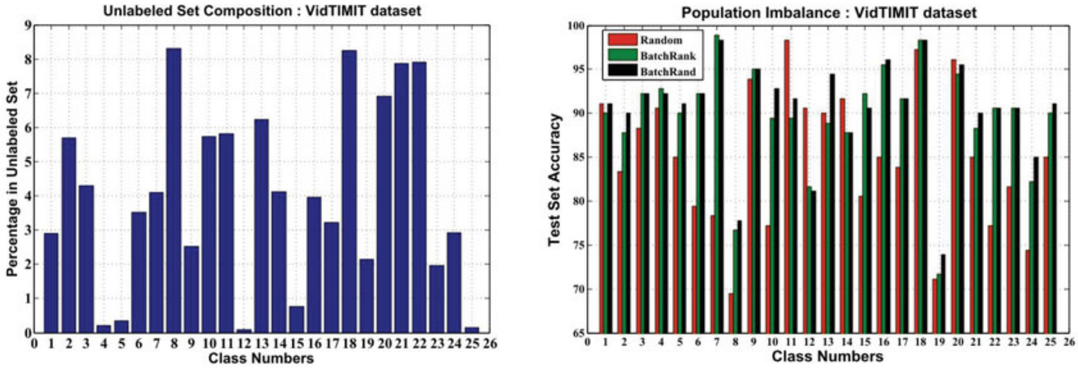
Random selection suffers since it does not integrate the composition of the unlabeled pool relative to the current



(a) 10 percent label noise

(b) 20 percent label noise

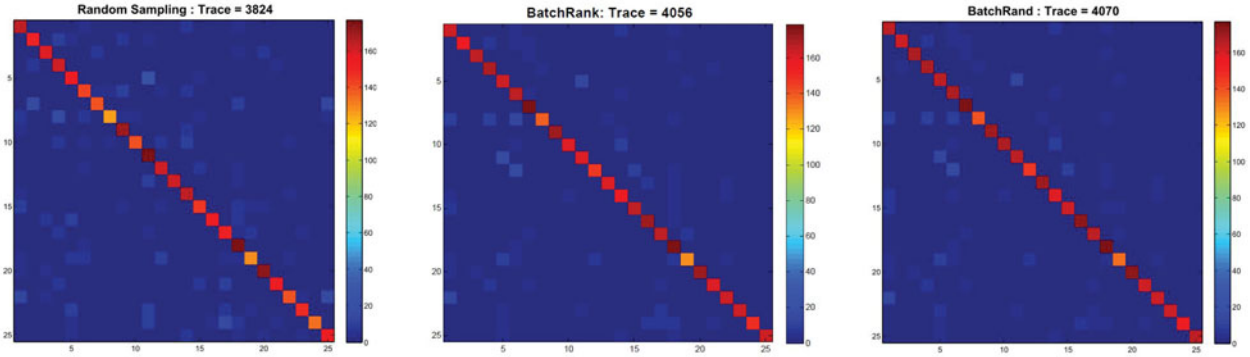
Fig. 5. Noise sensitivity of batchrank and batchrand on the VidTIMIT dataset. (Figures best viewed in color.)



(a) Percentage of images of each class in the unlabeled set

(b) Classwise Accuracy

Fig. 6. Class imbalance: VidTIMIT dataset. (Figures best viewed in color.)



(a) Random Selection

(b) BatchRank

(c) BatchRand

Fig. 7. Class imbalance: Confusion matrices for random selection, batchrank, and batchrand (Max trace = 4,500): VidTIMIT dataset.

training set in the batch selection process. In contrast, the proposed frameworks accurately identify the salient instances from the unlabeled set based on the uncertainty and divergence criteria. Thus, regardless of the composition of the unlabeled set, they append useful information to the underlying classification model and consistently deliver high accuracy on the individual classes. This is evident from Fig. 6b which plots the accuracy of each class for random sampling and the active learning methods. We see that the proposed algorithms depict better performance than random selection in 20 of the 25 classes. They also comprehensively outperform random sampling in the overall classification accuracy, as evident from the confusion matrices in Fig. 7. We note that BatchRank and BatchRand lead to much lower confusion among the classes and outperform random sampling by about 6 percent. This emphasizes their robustness to class imbalance.

7 Conclusion

In this paper, we proposed two novel batch mode active learning algorithms called BatchRank and BatchRand. Starting with an NP-hard optimization problem, we derived two convex relaxations and established bounds on the solution quality of each relaxation. Our empirical results on several challenging binary, multi-class and multi-label datasets corroborate the fact that the proposed approaches perform at par with the state-of-the-art BMAL techniques and also deliver high quality solutions. Our results also verified the proved theoretical bounds. We further demonstrated the robustness of the proposed algorithms to real-world issues like label noise and population imbalance. One of our future work will be to focus on the theoretical analysis of the algorithms where the weight matrix D can have negative entries. We will also consider extending this work to several other problems including binary matrix factorization and transfer-active learning.

Acknowledgments

S. Chakraborty is the corresponding author.

References

[1] S. Hoi, R. Jin, and M. Lyu, “Batch mode active learning with applications to text categorization and image retrieval,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1233–1248, Sep. 2009.

[2] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu, “Batch mode active learning and its application to medical image classification,” in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 417–424.

[3] S. C. H. Hoi, R. Jin, and M. R. Lyu, “Large-scale text categorization by batch mode active learning,” in *Proc. 15th Int. Conf. World Wide Web*, 2006, pp. 633–642.

[4] Y. Guo and D. Schuurmans, “Discriminative batch mode active learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 593–600.

[5] Y. Guo, “Active instance sampling via matrix partition,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 802–810.

[6] A. Beygelzimer, S. Dasgupta, and J. Langford, “Importance weighted active learning,” in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 49–56.

[7] S. Hanneke, “A bound on the label complexity of agnostic active learning,” in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 353–360.

[8] M. Balcan, S. Hanneke, and J. Vaughan, "The true sample complexity of active learning," *Mach. Learn.*, vol. 80, pp. 111–139, 2010.

[9] B. Settles, "Active learning literature survey," Comput. Sci. Dept., Univ. Wisconsin-Masdison, Madison, WI, USA, Tech. Rep. 1648, 2010.

[10] V. Balasubramanian, S. Chakraborty, and S. Panchanathan, "Generalized query by transduction for online active learning," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshop*, 2009, pp. 1378–1385.

[11] S. Ho and H. Wechsler, "Query by transduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1557–1571, Sep. 2008.

[12] C. Monteleoni and M. Kaariainen, "Practical online active learning for classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.

[13] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learn. Res.*, vol. 2, pp. 45–66, 2000.

[14] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proc. 9th ACM Int. Conf. Multimedia*, 2001, pp. 107–118.

[15] D. Cohn, Z. Ghahramani, and M. Jordan, "Active learning with statistical models," *J. Artif. Intell. Res.*, vol. 4, pp. 129–145, 1996.

[16] Y. Freund, S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *J. Mach. Learn.*, vol. 28, pp. 133–168, 1997.

[17] R. Liere and P. Tadepalli, "Active learning with committees for text categorization," in *Proc. 14th Nat. Conf. Artif. Intell.*, 1997, pp. 591–596.

[18] Y. Guo and R. Greiner, "Optimistic active learning using mutual information," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2007, pp. 823–829.

[19] Y. Baram, R. El-Yaniv, and K. Luz, "Online choice of active learning algorithms," *J. Mach. Learn. Res.*, vol. 5, pp. 255–291, 2004.

[20] A. McCallum and K. Nigam, "Employing EM and pool-based active learning for text classification," in *Proc. 15th Int. Conf. Mach. Learn.*, 1998, pp. 350–358.

[21] S. Huang, R. Jin, and Z. Zhou, "Active learning by querying informative and representative examples," in *Proc. Neural Inf. Process. Syst.*, 2010, pp. 892–900.

[22] M. Bilgic, L. Mihalkova, and L. Getoor, "Active learning for networked data," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 79–86.

[23] A. Harpale and Y. Yang, "Active learning for multi-task adaptive filtering," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 431–438.

[24] K. Brinker, "Incorporating diversity in active learning with support vector machines," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 59–66.

[25] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 839–846.

[26] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang, "Representative sampling for text classification using support vector machines," in *Proc. 25th Eur. Conf. Inf. Retrieval Res.*, 2003, pp. 393–407.

[27] S. Hoi, R. Jin, J. Zhu, and M. Lyu, "Semi-supervised SVM batch mode active learning for image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–7.

[28] S. Chakraborty, V. Balasubramanian, and S. Panchanathan, "Generalized batch mode active learning for face-based biometric recognition," *J. Pattern Recog.*, vol. 46, pp. 497–508, 2012.

[29] M. Kukar, "Transductive reliability estimation for medical diagnosis," *J. Artif. Intell. Med.*, vol. 29, pp. 81–106, 2003.

[30] S. Chakraborty, V. Balasubramanian, and S. Panchanathan, "Optimal batch selection for active learning in multi-label classification," in *Pro. 19th ACM Int. Conf. Multimedia*, 2011, pp. 1413–1416.

[31] X. Yuan and T. Zhang, "Truncated power method for sparse eigenvalue problems," *J. Mach. Learn. Res.*, vol. 14, pp. 899–925, 2013.

[32] M. Goemans and D. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, pp. 1115–1145, 1995.

[33] C. Sanderson, *Biometric Person Recognition: Face, Speech and Fusion*. Saarbrücken, Germany: VDM Verlag, Jun. 2008.

[34] S. Marcel, C. McCool, and P. Matejka, "On the results of the first mobile biometry (MOBIO) face and speaker verification evaluation," *Recognizing Patterns in Signals, Speech, Images and Videos, Lecture Notes in Computer Science*, vol. 6388, pp. 210–225, 2010.

[35] H. Ekenel, M. Fischer, Q. Jin, and R. Stiefelhagen, "Multi-modal person identification in a smart environment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.

[36] M. Pantic, M. Valstar, R. Rademaker, and L. Maat, "Web-based database for facial expression analysis," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2005, pp. 317–321.

[37] R. El-Kaliouby and P. Robinson, "Mind reading machines: Automated inference of cognitive mental states from video," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2004, pp. 682–688.

[38] P. Viola and M. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, pp. 137–154, 2004.

[39] G. Littlewort, M. Bartlett, I. Fasel, J. Susskind, and J. Movellan, "Dynamics of facial expression extracted automatically from video," *J. Image Vis. Comput.*, vol. 24, pp. 615–625, 2006.

[40] M. Zhang and K. Zhang, "Multi-label learning by exploiting label dependency," in *Proc. 10th ACM Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 999–1008.

[41] K. Brinker, "On active learning in multi-label classification," *Data and Information Analysis to Knowledge Engineering*, pp. 206–213, 2006.

[42] M. Singh, E. Curran, and P. Cunningham, "Active learning for multi-label image annotation," School Comput Sci. Informat., Univ. College Dublin, Dublin, Ireland, Tech. Rep. UCD-CSI-2009-01, 2009.

Shayok Chakraborty received the PhD degree in computer science from Arizona State University (ASU) in April 2013 under the mentorship of Dr. Sethuraman (Panch) Panchanathan. He was with IBM India as an application developer for one year until 2007. From July 2013 to June 2014, he was a postdoctoral research associate at Intel Labs, Oregon. He is currently a postdoctoral researcher in the Electrical and Computer Engineering (ECE) Department at Carnegie Mellon University. His research interests include machine learning, computer vision, data mining, and assistive technology. His PhD dissertation on batch mode active learning was nominated for the best PhD Dissertation Award in the Computer Science Department at ASU. He was featured as an outstanding graduate student by the Graduate College at Arizona State University and was selected for a research internship in the Machine Learning Department at Microsoft Research, Redmond. He has published his research in reputed conferences and journals in the areas of computer vision, data mining, and machine learning. He has also delivered a tutorial on active learning at the ICME 2013 conference. He is a member of the IEEE.

Vineeth Balasubramanian received dual master's degrees in mathematics and computer science from the Sri Sathya Sai Institute of Higher Learning, India, in 2001 and 2003, respectively, and was with the Oracle Corporation for two years until 2005. Until July 2013, he was an assistant research professor at the Center for Cognitive Ubiquitous Computing (CUbiC) at Arizona State University (ASU). He is currently an assistant professor in the Department of Computer Science and Engineering at the Indian Institute of Technology, Hyderabad, India. His PhD dissertation (2010) on the Conformal Predictions framework was nominated for the Outstanding PhD Dissertation at the Department of Computer Science at ASU. He received the Gold Medals for Academic Excellence in the Bachelors Program in Math in 1999 and the Masters Program in Computer Science in 2003. His research interests include pattern recognition, machine learning, computer vision, and multimedia computing within assistive and healthcare applications. He has more than 40 research publications in premier peer-reviewed venues, three patents under review, and received research grants from the US National Science Foundation in these fields. He is a member of the IEEE, the ACM, and the AAAI.

Qian Sun received the BS degree in electrical engineering and automation from the Nanjing University of Aeronautics and Astronautics, China, in 2008. She is currently working toward the PhD degree in computer science at Arizona State University. Her research interests include data mining and machine learning with the applications in bioinformatics. She is a member of the IEEE.

Sethuraman Panchanathan is currently the senior vice president in the Office of Knowledge Enterprise Development at Arizona State University. He is a foundation chair in computing and informatics and the director of the Center for Cognitive Ubiquitous Computing (CUbiC). He was the founding director of the School of Computing and Informatics and was instrumental in founding the Biomedical Informatics Department at ASU. He was the chair of the Computer Science and Engineering Department. His research interests include ubiquitous computing environments for enhancing quality of life for individuals with disabilities, haptic user interfaces, face/gait analysis and recognition, medical image processing, media processor design, and human-centered multimedia computing. He has published more than 430 research papers and articles. He has mentored more than 125 graduate students, post-docs, research engineers, and research scientists. He was appointed to the National Science Board in June 2014 by President Barack Obama. He is a member of the National Academy of Inventors and the Canadian National Academy of Engineering. He is a fellow of the IEEE and the SPIE.

Jieping Ye received the PhD degree in computer science from the University of Minnesota, Twin Cities, in 2005. He is an associate professor of computer science and engineering at Arizona State University. He is a core faculty member of the Bio-design Institute at ASU. His research interests include machine learning, data mining, and biomedical informatics. He was a senior program committee/area chair/program committee vice chair of many conferences including NIPS, KDD, IJCAI, ICDM, SDM, ACML, and PAKDD. He is an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence*. He received the SCI Young Investigator of the Year Award at ASU in 2007, the SCI Researcher of the Year Award at ASU in 2009, and the NSF CAREER Award in 2010. His papers have been selected for the outstanding student paper at the International Conference on Machine Learning in 2004, the KDD best research paper honorable mention in 2010, the KDD best research paper nomination in 2011 and 2012, the SDM best research paper runner up in 2013, and the KDD best research paper runner up in 2013. He is a senior member of the IEEE.