Batch Mode Active Learning with Applications to Text Categorization and Image Retrieval

Steven C.H. Hoi, *Member, IEEE,* Rong Jin, *Member, IEEE,* and Michael R. Lyu, *Fellow, IEEE*

Abstract— **Most machine learning tasks in data classification and information retrieval require human to provide labeled data examples in the training stage. The goal of active learning is to select the most informative examples for manual labeling in these learning tasks. Most of the previous studies in active learning have focused on selecting a *single* unlabeled example in each iteration. This could be inefficient, since the classification model has to be retrained for every acquired labeled example. It is also inappropriate for the setup of information retrieval tasks where the user's relevance feedback is often provided for the top K retrieved items. In this paper, we present a framework for batch mode active learning, which selects a number of informative examples for manual labeling in each iteration. The key feature of batch mode active learning is to reduce the redundancy among the selected examples such that each example provides unique information for model updating. To this end, we employ the Fisher information matrix as the measurement of model uncertainty, and choose the set of unlabeled examples that can efficiently reduce the Fisher information of the classification model. We apply our batch mode active learning framework to both text categorization and image retrieval. Promising results show that our algorithms are significantly more effective than the active learning approaches that select unlabeled examples based only on their informativeness for the classification model.**

Index Terms— **Active Learning, Batch Mode Active Learning, Logistic Regressions, Kernel Logistic Regressions, Convex Optimization, Text Categorization, Image Retrieval**

I. Introduction

Data classification has been an active research topic in the machine learning community for many years. The goal of data classification is to automatically assign data examples to a set of predefined categories. One prerequisite for any data classification scheme is to have labeled examples. In order to reduce the effort in acquiring labeled examples, a number of active learning methods [6], [7], [28], [4], [27], [35] have been developed for data classification. The key idea of active learning is to identify the examples that are most informative with respect to the current classification model. In the past, active learning has been successfully applied to a number of applications, including text categorization [24], [25], [35], computer vision [20], content-based image retrieval (CBIR) [31], [11], and text document retrieval [29].

Most active learning algorithms are conducted in an iterative fashion. In each iteration, the example with the highest classification uncertainty is chosen for manual labeling, and the classification model is retrained with the additional labeled example. The step of training a classification model and the step of soliciting a label are iterated alternately until most of the unlabeled examples can be classified with reasonably high confidence. A main problem with such a scheme is that only a *single* example is selected for labeling in each iteration. As a result, the classification model has to be retrained after each new example is labeled. In the paper, we propose a novel active learning framework that is able to select a *batch* of unlabeled examples simultaneously in each iteration. A simple strategy toward the *batch mode active learning* is to select the k most informative examples. The problem with such an approach is that some of the selected examples could be similar, or even identical, to each other, and therefore do not provide additional information for model updating. In general, the key of batch mode active learning is to ensure little redundancy among the selected examples such that each example provides unique information for model updating.

To this end, we propose a framework of batch mode active learning that measures the overall information for a set of unlabeled examples by the Fisher information matrix [10]. We formulate the batch mode active learning framework into a Semi-Definite Programming (SDP) problem, and present an effective optimization algorithm based on the bound optimization technique. Further, we propose the kernel version of the proposed technique for kernel logistic regression models. Finally, we present an empirical study of the proposed batch mode active learning algorithm for two real world applications, i.e., text categorization and content-based image retrieval.

The rest of this paper is organized as follows. Section II reviews related work on active learning, text categorization, and image retrieval. Section III briefly introduces the concepts of logistic regression and kernel logistic regressions, which are used as the classification model in our study. Section IV presents the framework of batch mode active learning and an efficient algorithm for solving the related optimization problem. Sections V and VI present the empirical study of batch mode active learning for text categorization and content-based image retrieval, respectively. Section VII gives an empirical evaluation of two different implementations of batch mode active learning using different optimization approaches. Section VIII sets out our conclusions.

A short version of this paper has been published in the proceedings of the 15th International World Wide Web Conference (WWW2006). Different from the conference paper, the new contents of this paper include (1) the kernel version of batch mode active learning, (2) an improved qualitative analysis that explains why the Fisher information matrix is a suitable approach for batch mode active learning, (3) a new extensive set of empirical study of batch mode active learning for classification, and (4) the empirical comparison between the two optimization formulations of batch mode active learning, i.e., the Semi-Definitive Programming formulation and the approximate formulation by bound optimization.

Steven C.H. Hoi is currently with the School of Computer Engineering, at the Nanyang Technological University. Rong Jin is with the department of Computer Science and Engineering, at the Michigan State University. Michael R. Lyu is with the department of Computer Science and Engineering, at the Chinese University of Hong Kong. E-mail: chhoi@ntu.edu.sg, lyu@cse.cuhk.edu.hk, rongjin@cse.msu.edu, Tel: (+65) 6513-8040, Fax: (+65) 6792-6559.

II. Related Work

We will first review related work on active learning, and then discuss text categorization and content-based image retrieval.

A. Active Learning

Active learning, or so-called pool-based active learning, has been extensively studied in machine learning for a number of years, and has already been employed for text categorization and image retrieval in the past [18], [19], [24], [25], [34]. Most active learning algorithms are conducted in an iterative fashion, alternating between updating the classification models and soliciting class labels for the most informative examples. One of the key issues in active learning is how to measure the classification uncertainty of unlabeled examples. The ensemble based approaches [6], [7], [24], [28] measure the classification uncertainty based on the predictions by an ensemble of classification models. They first generate a number of distinct classification models using the labeled examples; then, the classification uncertainty of a test example is measured by the amount of disagreement among the ensemble of classification models in predicting the labels for the test example. Another group of approaches measure the classification uncertainty of a test example by how far the example is away from the classification boundary (i.e., classification margin) [4], [27], [35]. A well-known approach within this group is *Support Vector Machine Active Learning* developed by Tong and Koller [35]. Due to its popularity and success in previous studies, we will use it as the baseline approach in our empirical study.

B. Text Categorization

The first application in our study is text categorization. Text categorization has been widely studied in the communities of data mining, information retrieval and statistical learning [39], [40]. More recently, text categorization techniques have been the key toward automated categorization of web pages and web sites, which is being further applied to improve the performance of web search engines in finding relevant documents and facilitating users in browsing web pages or web sites.

In the past decade, a large number of statistical learning techniques have been applied to automatic text categorization [39], including the K-Nearest Neighbor approaches [23], decision trees [2], Bayesian classifiers [36], inductive rule learning [5], neural networks [26], and support vector machines (SVM) [13]. Empirical studies in recent years [13], [39] have shown that SVM is one of the state-of-the-art techniques among the methods mentioned above.

Recently, logistic regression has attracted considerable attention for text categorization and high-dimension data mining [16]. Several recent studies have shown that the logistic regression model can achieve comparable classification accuracy to SVMs in text categorization. Compared to SVMs, the logistic regression model is usually more efficient in model training, especially when the number of training documents is large [17]. Furthermore, the posterior probability output by the logistic regression model can be used as the intermediate results for other models, such as the Hierarchical Mixture Expert (HME) model [14]. This motivates us to use logistic regression as the basis classifier for text categorization.

One critical issue for automated text categorization is how to reduce the number of labeled documents that are required for building reliable text classification models. Given the substantial effort required to acquire labels for documents, the key is to exploit the unlabeled documents. One solution is the semi-supervised learning approach, which tries to learn a classification model from a mixture of labeled and unlabeled documents. A comprehensive study of semi-supervised learning techniques can be found in [44]. Another solution is active learning [22], [28], which tries to choose the most informative examples for manual labeling. In this paper, we focus our attention on using active learning for reducing the effort required for manual labeling.

C. Image Retrieval

The second application in our study is content-based image retrieval (CBIR). One of the key challenges in CBIR is the semantic gap between the low-level visual features that are used to represent images and the high-level semantic concepts that are conveyed in the content of images. One popular approach in CBIR toward bridging the semantic gap is relevance feedback, in which a classification model is learned from the user's relevance judgments on the top retrieved images. During the past years, a variety of machine learning algorithms have been proposed for relevance feedback, including Bayesian learning [38], decision tree [21], boosting [33], discriminant analysis [11], and support vector machines [42], [9], etc. Among them, the kernel based classifiers, such as support vector machines, have shown to be one of the promising approaches for relevance feedback [9].

A typical approach for relevance feedback of CBIR will first rank the images according to their probability of being classified as similar to the query example, and solicit relevance judgments on the top ranked images from the users. The acquired labeled images will then be used to update the classification model. The problem with such an approach is that the most similar images identified by a classification model may not be informative with respect to the classification model. Recently, active learning has been suggested as a more promising approach for soliciting users' relevance feedback. One of the most popular approaches may be the support vector machine active learning [34], which solicits users' relevance judgments for the images that are closest to the decision boundary of the classification model. However, directly applying active learning methods to relevance feedback is insufficient given that most active learning methods can only identify the single most informative example, while the relevance feedback of CBIR usually solicits the relevance judgments on multiple images. Although the authors in [34] presented a simple and efficient batch sampling solution, the heuristics are not well justified and depend on the context of the problems. In contrast, the proposed batch mode active learning algorithm is well founded on the basis of Fisher information. Furthermore, we presented a bound optimization algorithm that solves the related optimization problem efficiently.

III. Logistic Regression and Kernel Logistic Regression

In this section, we give a brief introduction to logistic regression and kernel logistic regression, which are used as the basis classification models in text categorization and content-based image retrieval, respectively.

A. Logistic Regression

Logistic regression (LR) is a binary class classification model, and has been widely used in data mining and machine learning due to its close relations to Support Vector Machines and Adaboost [37], [41].

Given the input features $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ of a test example where d is the number of features, logistic regression models the conditional probability of assigning a class label y to the example by

$$p(y|\mathbf{x}) = \frac{1}{1+\exp(-y(\mathbf{w}^\top \mathbf{x} + b))} \qquad (1)$$

where $y \in \{+1, -1\}$ is the class label, $\mathbf{w} = (w_1, w_2, \ldots, w_d)$ are the weights assigned to input features, and b is the bias term. In general, logistic regression is a linear classifier that has been shown to be very effective in classifying text documents [16]. In addition, a number of efficient algorithms have appeared in the recent literature [17] that allow logistic regression to handle large-scale text categorization problems effectively.

B. Kernel Logistic Regression

Kernel logistic regression (KLR) is a nonlinear extension of the traditional logistic regression model based on the kernel machine theory [37]. More specifically, given the training examples $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$ and a kernel function $K(\cdot,\cdot)$, the kernel logistic regression is posed as the following optimization problem:

$$\min_{f \in \mathcal{H}_K} \frac{1}{n}\sum_{i=1}^{n} \ln(1 + e^{-y_i f(\mathbf{x}_i)}) + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}_K} \qquad (2)$$

where $\mathcal{H}_K$ is the Hilbert space reproduced by the kernel function K. According to the representer theorem [15], the optimal $f(\mathbf{x})$ can be written in the following form:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i) \qquad (3)$$

In the above, we omit the bias term b in $f(x)$ for simplicity. Using the parametric form of $f(x)$, the problem in Eq. 2 is converted into an optimization problem of dual variables α.

It is interesting to note the close relationship between KLR and kernel SVM [8]. To see this, we write the kernel SVM into the following form:

$$\min_{f \in \mathcal{H}_K} \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i f(\mathbf{x}_i)) + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}_K} \qquad (4)$$

Comparing the above equation to Eq. 2, we see that both problems share a similar form, i.e., *loss* + *penalty*. The key difference between these two algorithms lies in the loss functions. According to a number of studies [8], [45], KLR achieves a comparable classification accuracy to kernel SVM, and enjoys several important merits, such as natural probability outputs.

IV. A Framework of Batch Mode Active Learning

In this section, we present a framework of batch mode active learning for data classification tasks. In our proposed scheme, logistic regression is used as the underlying classification model for the binary classification tasks. In the following subsections, we first introduce theoretic foundation for the proposed framework, including the Fisher information matrix and its application to active learning. Using the theoretic foundation, we present a framework of batch mode active learning, with a qualitative analysis aiming to illustrate how the optimization of Fisher information will eliminate the overlap among the selected examples in active learning. Finally, we present two algorithms to efficiently solve the optimization problem related to batch mode active learning.

A. Theoretical Foundation

Our active learning methodology is motivated by the work in [43], in which the authors presented a theoretical framework of active learning based on the minimization of Fisher information. Given a data distribution $q(x)$, and a classification model $p(y|x;\alpha)$ where α includes all the parameters of the classification model, the Fisher information matrix is defined as follows

$$I_q(\alpha) = -\int q(x)dx \int p(y|\alpha, x)\frac{\partial^2}{\partial \alpha^2} \ln p(y|x;\alpha)dy \qquad (5)$$

For the logistic regression model, its Fisher information matrix $I_q(\alpha)$ is attained as:

$$\begin{aligned} I_q(\alpha) &= -\int q(\mathbf{x}) \sum_{y=\pm 1} p(y|\mathbf{x}) \frac{\partial^2}{\partial \alpha^2} \log p(y|\mathbf{x}) d\mathbf{x} \\ &= \int \frac{1}{1+\exp(\alpha^\top \mathbf{x})} \frac{1}{1+\exp(-\alpha^\top \mathbf{x})} \mathbf{x}\mathbf{x}^\top q(\mathbf{x}) d\mathbf{x} \qquad (6) \end{aligned}$$

Fisher Information matrix, is widely used in statistics for measuring model uncertainty [30]. The most well known result is the Cramer-Rao bound. Since the objective of active learning is to identify examples that are most informative to the target classification model, we will select examples that can effectively reduce the Fisher information of the classification model, which forms the basis for the active learning framework presented in [43]. More specific, we denote by $p(\mathbf{x})$ the distribution of all unlabeled examples, and by $q(\mathbf{x})$ the distribution of unlabeled examples that are chosen for manual labeling. Let α denote the parameters of a classification model. Let $I_p(\alpha)$ and $I_q(\alpha)$ denote the Fisher information matrix of the classification model for the distribution $p(\mathbf{x})$ and $q(\mathbf{x})$, respectively. Then, the set of examples that can most efficiently reduce the uncertainty of the classification model is found by minimizing the ratio between the two Fisher information matrices $I_p(\alpha)$ and $I_q(\alpha)$, i.e.,

$$q^* \;=\; \arg\min_q \mathrm{tr}(I_q(\alpha)^{-1} I_p(\alpha)) \qquad (7)$$

B. Problem Formulation

Let $D = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ be the unlabeled data, and $S = (\mathbf{x}_1^s, \mathbf{x}_2^s, \ldots, \mathbf{x}_k^s)$ be the subset of selected examples, where k is the number of examples to be selected. In order to estimate the optimal distribution $q(\mathbf{x})$, we replace the integration in Eqn. (6) with the summation over the unlabeled data, and the model parameter α with its empirical estimation $\hat{\alpha}$.

We can now rewrite the above expression for the two Fisher information matrices I_p and I_q as:

$$I_p(\hat{\alpha}) = \frac{1}{n}\sum_{\mathbf{x}\in D}\pi(\mathbf{x})(1-\pi(\mathbf{x}))\mathbf{x}\mathbf{x}^\top + \delta I_d$$

$$I_q(S,\hat{\alpha}) = \frac{1}{k}\sum_{\mathbf{x}\in S}\pi(\mathbf{x})(1-\pi(\mathbf{x}))\mathbf{x}\mathbf{x}^\top + \delta I_d$$

where

$$\pi(\mathbf{x}) = p(-|\mathbf{x}) = \frac{1}{1+\exp(\hat{\alpha}^\top\mathbf{x})} \tag{8}$$

In the above, $\hat{\alpha}$ stands for the classification model that is estimated from the labeled examples. I_d is the identity matrix of size $d\times d$. δ is the smoothing parameter. δI_d is added to the estimation of $I_p(\hat{\alpha})$ and $I_q(S,\hat{\alpha})$ to prevent them from being singular matrices. Hence, the final optimization problem for batch mode active learning is formulated as follows:

$$S^* \quad = \quad \mathop{\arg\min}_{S\subseteq D\wedge|S|=k} \operatorname{tr}(I_q(S,\hat{\alpha})^{-1}I_p(\hat{\alpha})) \tag{9}$$

C. Qualitative Analysis

In this section, we will qualitatively justify the theory of minimizing the ratio of Fisher information for batch mode active learning. In particular, we consider two cases, the case of selecting a single unlabeled example and the case of selecting multiple unlabeled examples simultaneously. To simplify our discussion, we assume $\|\mathbf{x}_i\|_2^2 = 1$ for any unlabeled example $\mathbf{x}_i$.

a) Selecting a single unlabeled example: The Fisher information matrix I_q is simplified into the following form when the i-th example is selected:

$$I_q(\hat{\alpha};\mathbf{x}_i) \quad = \quad \pi_i(1-\pi_i)\mathbf{x}_i\mathbf{x}_i^\top + \delta I_d$$

Note that the above matrix has eigenvalue $\pi_i(1-\pi_i)+\delta$ for eigenvector $\mathbf{x}_i$ and δ for other eigenvectors. Thus, the objective function $\operatorname{tr}(I_q(\hat{\alpha})^{-1}I_p(\hat{\alpha}))$ becomes:

$$\operatorname{tr}(I_q(\hat{\alpha})^{-1}I_p(\hat{\alpha})) = \frac{1}{n\delta}\sum_{j=1}^n \pi_j(1-\pi_j)$$
$$-\frac{\pi_i(1-\pi_i)}{n\delta(\delta+\pi_i(1-\pi_i))}\sum_{j=1}^n \pi_j(1-\pi_j)(\mathbf{x}_i^\top x_j)^2$$

As indicated by the above expression, to minimize the above expression, we need to maximize (1) $\pi_i(1-\pi_i)$, and (2) $\mathbf{x}_i^\top\mathbf{x}_j, \forall j \neq i$. Since $\pi_i(1-\pi_i)$ reaches its maximum value at $\pi_i = 0.5$, it can be regarded as the measurement of classification uncertainty for the i-th unlabeled example. Thus, the optimal example chosen by minimizing the ratio of Fisher information matrix in the above expression tends to be the one with a high classification uncertainty. Furthermore, the quantity $\mathbf{x}_i^\top\mathbf{x}_j, \forall j \neq i$ measures the similarity of the ith example to the remaining unlabeled examples. Thus, by maximizing $\mathbf{x}_i^\top\mathbf{x}_j, \forall j \neq i$, the selected example tends to be representative of the entire collection of unlabeled examples.

b) Selecting multiple unlabeled examples simultaneously: Let $\mathcal{S} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k)$ be the k ($k>1$) selected examples. Then, the Fisher information matrix $I_q(\hat{\alpha};\mathcal{S})$ is written as

$$\begin{aligned} I_q(\hat{\alpha};\mathcal{S}) &= \frac{1}{k}\sum_{i=1}^k \pi_i(1-\pi_i)\mathbf{x}_i\mathbf{x}_i^\top + \delta I_d \\ &= \frac{1}{k}\left(\sum_{i=1}^k \pi_i(1-\pi_i)\right)\left(\mathbf{m}^\top\mathbf{m}+\Lambda\right) + \delta I_d \end{aligned}$$

where $\mathbf{m}$ and Λ is the mean and the covariance matrices defined as follows:

$$\begin{aligned} \mathbf{m} &= \sum_{i=1}^k \frac{\pi_i(1-\pi_i)}{\sum_{j=1}^k \pi_j(1-\pi_j)}\mathbf{x}_i \\ \Lambda &= \sum_{i=1}^k \frac{\pi_i(1-\pi_i)}{\sum_{j=1}^k \pi_j(1-\pi_j)}(\mathbf{x}_i-\mathbf{m})(\mathbf{x}_i-\mathbf{m})^\top \end{aligned}$$

First, note that the covariance matrix Λ is a positive definite matrix with rank equal to $k-1$ if we assume that all the unlabeled examples are linear independent and furthermore $d > k$. Second, if $(\lambda_i, \mathbf{v}_i), i = 1, 2, \ldots, k-1$ are the eigenvalues and the eigenvectors of Λ, we have

$$\mathbf{m}^\top\mathbf{v}_i = 0, i = 1, 2, \ldots, k-1$$

This is because $\mathbf{m}^\top\Lambda\mathbf{m} = 0$. Based on these two observations, we have $\operatorname{tr}(I_q(\hat{\alpha};\mathcal{S})^{-1}I_p(\hat{\alpha}))$ approximated as:

$$\begin{aligned} &\operatorname{tr}(I_q(\hat{\alpha};\mathcal{S})^{-1}I_p(\hat{\alpha})) \\ \approx\ & \frac{k}{n\sum_{i=1}^k \pi_i(1-\pi_i)}\sum_{j=1}^n \pi_j(1-\pi_j)\left((\mathbf{m}^\top\mathbf{x}_j)^2 + \mathbf{x}_j\Lambda^\dagger\mathbf{x}_j\right) \\ +\ & \frac{1}{n\delta}\sum_{j=1}^n \pi_j(1-\pi_j)\left(1 - \frac{(\mathbf{m}^\top\mathbf{x}_j)^2}{\|\mathbf{m}\|_2^2} - \sum_{i=1}^{k-1}\lambda_i(\mathbf{v}_i\mathbf{x}_j)^2\right) \end{aligned} \tag{10}$$

where $\dagger$ stands for pseudo inverse. In the above, we assume that $\delta \ll \sum_{i=1}^k \pi_i(1-\pi_i)/k$. As indicated by the above expression, to minimize the ratio between $I_q(\hat{\alpha};\mathcal{S})$ and $I_p(\hat{\alpha})$, we need to find a set of examples that satisfy the following conditions:

- The selected examples should have large $\sum_{i=1}^k \pi_i(1-\pi_i)$. This implies that all the selected examples should have large classification uncertainty.
- The selected examples should have a large covariance matrix Λ such that $\mathbf{x}_j^\top\Lambda^\dagger\mathbf{x}_j$ is small for any unlabeled example. This implies that the selected examples should be diverse enough such that their covariance matrix Λ provides a good description for the distribution of all the unlabeled examples.

Thus, by minimizing the Fisher information matrix, we can avoid choosing the examples that are similar to each other.

D. Batch Mode Active Learning via Semi-Definitive Programming (SDP)

It is not easy to find an appropriate distribution q(x) that minimizes $\mathbf{tr}(I_q^{-1}I_p)$. In the following, we present the semidefinite programming (SDP) approach for optimizing $\mathbf{tr}(I_q^{-1}I_p)$.

The key challenge in solving the problem in (7) is that the Fisher information matrix for the selected examples I_q is presented in the form of matrix inverse in (7). As a result, the objective function is a nonlinear function for the selected examples. Below, we aim to linearize the optimization problem in (7). To this end, we rewrite the objective function $\mathbf{tr}(I_q^{-1}I_p)$ as $\mathbf{tr}(I_p^{1/2}I_q^{-1}I_p^{1/2})$, and introduce a slack matrix $M \in \mathbf{R}^{n\times n}$ to upper bound the objective function, i.e., $M \succeq I_p^{1/2}I_q^{-1}I_p^{1/2}$. Then the original optimization problem can be rewritten as follows:

$$
\begin{aligned}
\min_{\mathbf{q},M} \quad & \mathbf{tr}(M) \\
\text{s. t.} \quad & M \succeq I_p^{1/2}I_q^{-1}I_p^{1/2} \\
& \sum_{i=1}^n q_i = 1, q_i \geq 0, i = 1,\ldots,n
\end{aligned}
\tag{11}
$$

In the above, we use the property $\mathbf{tr}(A) \geq \mathbf{tr}(B)$ if $A \succeq B$. Furthermore, we use the Schur complementary [3], i.e.,

$$
D \succeq AB^{-1}A^\top \Leftrightarrow \begin{pmatrix} B & A^\top \\ A & D \end{pmatrix} \succeq 0 \tag{12}
$$

if $B \succeq 0$. This will lead to the following formulation for the problem in (11):

$$
\begin{aligned}
\min_{\mathbf{q},M} \quad & \mathbf{tr}(M) \\
\text{s. t.} \quad & \begin{pmatrix} I_q & I_p^{1/2} \\ I_p^{1/2} & M \end{pmatrix} \succeq 0 \\
& \sum_{i=1}^n q_i = 1, q_i \geq 0, i = 1,\ldots,n
\end{aligned}
\tag{13}
$$

or more specifically

$$
\begin{aligned}
\min_{\mathbf{q},M} \quad & \mathbf{tr}(M) \\
\text{s. t.} \quad & \sum_{i=1}^n q_i \begin{pmatrix} \pi_i(1-\pi_i)\mathbf{x}_i\mathbf{x}_i^\top & I_p^{1/2} \\ I_p^{1/2} & M \end{pmatrix} \succeq 0 \\
& \sum_{i=1}^n q_i = 1, q_i \geq 0, i = 1,\ldots,n
\end{aligned}
\tag{14}
$$

It is clearly that the above problem is linear in M. In fact, it belongs to the family of semi-definite programming and can be solved by standard convex optimization packages such as SeDuMi [32].

E. Eigen Space Simplification

Although the formulation in (14) is mathematically sound, directly solving the optimization problem could be computationally expensive when the size of matrix M is large. In particular, the high computational cost arises from the linear matrix inequality (LMI) in (14). To reduce the computational complexity, we aim to simplify the LMI constraint into a set of linear inequality constraints y assuming certain parametric form for M. In particular, we assume that M is only expanded in the eigen space of matrix I_p. Let $\{(\lambda_1, \mathbf{v}_1), \ldots, (\lambda_s, \mathbf{v}_s)\}$ be the top s ($s \ll n$) eigen vectors of matrix I_p where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_s > 0$. We assume matrix M has the following form:

$$
M = \sum_{k=1}^s \gamma_k \mathbf{v}_k \mathbf{v}_k^\top \tag{15}
$$

where the combination parameters $\gamma_k \geq 0,\ k = 1,\ldots,s$. We rewrite the inequality for $M \succeq I_p^{1/2}I_q^{-1}I_p^{1/2}$ as $I_q \succeq I_p^{1/2}M^{-1}I_p^{1/2}$. Using the expression for M in (15), we have

$$
I_p^{1/2}M^{-1}I_p^{1/2} = \sum_{k=1}^s \gamma_k^{-1}\lambda_k \mathbf{v}_k\mathbf{v}_k^\top \tag{16}
$$

Given that the necessary condition for $I_q \succeq I_p^{1/2}M^{-1}I_p^{1/2}$ is

$$
\mathbf{v}^\top I_q \mathbf{v} \geq \mathbf{v}^\top I_p^{1/2}M^{-1}I_p^{1/2}\mathbf{v},\ \forall \mathbf{v} \in \mathbf{R}^d \ ,
$$

we have $\mathbf{v}_k^\top I_q \mathbf{v}_k \geq \gamma_k^{-1}\lambda_k$ for $k = 1,\ldots,s$. This necessary condition leads to the following constraints for γ_k:

$$
\gamma_k \geq \frac{\lambda_k}{\mathbf{v}_k^\top I_q \mathbf{v}_k} = \frac{\lambda_k}{\sum_{i=1}^n q_i\pi_i(1-\pi_i)(\mathbf{x}_i^\top\mathbf{v}_k)^2},\ k = 1,\ldots,s \tag{17}
$$

Meanwhile, the objective function in (14) can be expressed as

$$
\mathbf{tr}(M) = \sum_{k=1}^s \gamma_k \tag{18}
$$

By putting the above two expressions together, we approximate the SDP problem in (14) into the following optimization:

$$
\begin{aligned}
\min_{\mathbf{q}\in\mathbf{R}^n} \quad & \sum_{k=1}^s \frac{\lambda_k}{\sum_{i=1}^n q_i\pi_i(1-\pi_i)(\mathbf{x}_i^\top\mathbf{v}_k)^2} \\
\text{s.t.} \quad & \sum_{i=1}^n q_i = 1, q_i \geq 0, i = 1,\ldots,n
\end{aligned}
\tag{19}
$$

Note that the above optimization problem is a convex optimization problem since $f(x) = 1/x$ is convex when $x \geq 0$. Given this formulation in (19), we present a bound optimization algorithm for solving the above optimization problem.

F. Bound Optimization Algorithm

The main idea of bound optimization algorithm is to update the solution iteratively. In each iteration, we will first calculate the difference between the objective function of the current iteration and the objective function of the previous iteration. Then, by minimizing the upper bound of the difference, we find the solution of the current iteration.

Let $\mathbf{q}'$ and $\mathbf{q}$ denote the solutions obtained in two consecutive iterations, and let $\mathcal{L}(\mathbf{q})$ be the objective function in (19). Based on the proof given in Appendix A, we have the following expression:

$$
\begin{aligned}
\mathcal{L}(\mathbf{q}) &= \sum_{k=1}^s \frac{\lambda_k}{\sum_{i=1}^n q_i\pi_i(1-\pi_i)(\mathbf{x}_i^\top\mathbf{v}_k)^2} \\
&\leq \sum_{i=1}^n \frac{(q_i')^2}{q_i}\pi_i(1-\pi_i)\sum_{k=1}^s \frac{(\mathbf{x}_i^\top\mathbf{v}_k)^2\lambda_k}{\left(\sum_{j=1}^n q_j'\pi_j(1-\pi_j)(\mathbf{x}_j^\top\mathbf{v}_k)^2\right)^2}
\end{aligned}
\tag{20}
$$

Now, instead of optimizing the original objective function $\mathcal{L}(\mathbf{q})$, we can optimize its upper bound, which leads to the following simple updating equation:

$$
\begin{aligned}
q_i &\longleftarrow q_i'^2 \pi_i(1-\pi_i) \sum_{k=1}^{s} \frac{(\mathbf{x}_i^\top \mathbf{v}_k)^2 \lambda_k}{\left(\sum_{j=1}^{n} q_j' \pi_j (1-\pi_j)(\mathbf{x}_j^\top \mathbf{v}_k)^2\right)^2} \\
q_i &\longleftarrow \frac{q_i}{\sum_{j=1}^{n} q_j}
\end{aligned}
\tag{21}
$$

As with all bound optimization algorithms [3], this algorithm is guaranteed to converge to a local maximum. Since the original optimization problem in (19) is a convex optimization problem, the above updating procedure is guaranteed to converge to a global optimum. Fig. 1 shows the algorithm for batch mode active learning by bound optimization techniques.

Remark. It is interesting to examine the property of the solution obtained by the updating equation in (21). First, according to (21), the example with a large classification uncertainty (i.e., $\pi_i(1-\pi_i)$) will be assigned a large probability q_i. This is because q_i is proportional to $\pi_i(1-\pi_i)$, the classification uncertainty of the i-th unlabeled example. Second, according to (21), any example that is similar to many unlabeled examples is more likely to be selected. This is because q_i is proportional to the term $(\mathbf{x}_i^\top \mathbf{v}_k)^2$, the similarity of the i-th example to all the principal eigenvectors. This is consistent with our intuition that we should select the most informative and representative examples for active learning.

```
Algorithm 1 Batch Mode Active Learning with Logistic Regression ($\mathrm{LR_{BMAL}}$)
INPUT: L, U /* training data set, unlabeled data set */
        n, m, k /* unlabeled size, training size, batch size */
VARIABLES: $\mathbf{q}$, $I_p$ /* sampling probability, Fisher information matrix*/
PARAMETERS: $\delta$, $s$ /* regularization factor, number of top eigens */
OUTPUT: S /* a batch of selected unlabeled examples */
PROCEDURE:
initialize: S = $\emptyset$;   $q_i \leftarrow 1/n, i = 1,2,\ldots,n$;
$\alpha$ = LR_train(L);
$\pi_i = p(+|\mathbf{x}_i) = 1/(1+\exp(-\alpha^\top \mathbf{x}_i)), i = 1,2,\ldots,n$;
$I_p \leftarrow \frac{1}{n}\sum_{i=1}^{n} \pi_i(1-\pi_i)\mathbf{x}_i\mathbf{x}_i^\top + \delta I_d$;
($\mathbf{v}$, $\lambda$) = eig($I_p$, s); /* do eigen decomposition */
while (change in $\{q_i\} > \epsilon$) do
    $f_j \leftarrow \sum_{i=1}^{n} q_i\pi_i(1-\pi_i)(\mathbf{x}_i^\top \mathbf{v}_j)^2, j = 1,2,\ldots,s$;
    $q_i \leftarrow q_i^2\pi_i(1-\pi_i)\sum_{j=1}^{s}(\lambda_j \mathbf{x}_i^\top \mathbf{v}_j)^2/f_j^2$
    $q_i \leftarrow q_i/\sum_{j=1}^{n} q_j, i = 1,2,\ldots,n$;
end while
while (|S| $>=$ $k$) do
    $\mathbf{x}^* = \arg\max_{\mathbf{x}_i \in \mathsf{U}} q(\mathbf{x}_i)$;
    S $\leftarrow$ S $\cup$ $\{\mathbf{x}^*\}$;
    U $\leftarrow$ U - $\{\mathbf{x}^*\}$;
end while
return S;
```

Fig. 1. A bound optimization algorithm for batch mode active learning

G. Batch Mode Active Learning for Kernel Logistic Regression

To extend the above analysis to the nonlinear classification model, we follow the idea of the imported vector machine reported by [45]. More specifically, we introduce the mapping function $f : \mathbf{x} \rightarrow \phi(\mathbf{x})$, and the kernel function $K(\mathbf{x}', \mathbf{x}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}')$ that calculates the dot product of two examples in the mapped space. Then, according to the results described in Section III-B, we have

$$
p(y|\mathbf{x}) = \frac{1}{1+\exp(-yK(\mathbf{w},\mathbf{x}))}
$$

where

$$
K(\mathbf{w},\mathbf{x}) = \sum_{\mathbf{x}' \in L} \theta(\mathbf{x}')K(\mathbf{x}',\mathbf{x}),
$$

$L = ((y_1, \mathbf{x}_1^L), (y_2, \mathbf{x}_2^L), \ldots, (y_m, \mathbf{x}_m^L))$ represents the set of labeled examples, and $\theta(\mathbf{x})$ is the combination weight for labeled example $\mathbf{x}$. Thus, by treating $(K(\mathbf{x}_1^L, \mathbf{x}), K(\mathbf{x}_2^L, \mathbf{x}), \ldots, K(\mathbf{x}_m^L, \mathbf{x}))$ as the new representation for the unlabeled example $\mathbf{x}$, we can directly apply the result for the linear logistic regression model to the nonlinear case. Specifically, we can represent the Fisher information matrix I_p as follows:

$$
I_p(\hat{\alpha}) = \sum_{i=1}^{n} \pi_i(1-\pi)\mathbf{g}_i\mathbf{g}_i^\top \tag{22}
$$

where $\mathbf{g}_i = (K(\mathbf{x}_1^L, \mathbf{x}_i), K(\mathbf{x}_2^L, \mathbf{x}_i), \ldots, K(\mathbf{x}_m^L, \mathbf{x}_i))$ and $\pi_i = p(-|\mathbf{x}_i)$. Similarly, I_q can also be represented as:

$$
I_q(\hat{\alpha}) = \sum_{i=1}^{n} \pi_i(1-\pi_i)q_i\mathbf{g}_i\mathbf{g}_i^\top \tag{23}
$$

Hence, the convex optimization problem can be rewritten as:

$$
\begin{aligned}
\min_{\mathbf{q},M} \quad & \mathbf{tr}(M) \\
\text{s. t.} \quad & \sum_{i=1}^{n} q_i \begin{pmatrix} \pi_i(1-\pi_i)\mathbf{g}_i\mathbf{g}_i^\top & I_p^{1/2} \\ I_p^{1/2} & M \end{pmatrix} \succeq 0 \\
& \sum_{i=1}^{n} q_i = 1, q_i \geq 0, i = 1,\ldots,n
\end{aligned}
\tag{24}
$$

Finally, by using the similar approach in the above derivation, we can develop an algorithm of batch mode active learning with kernel logistic regressions shown in Fig. 2.

```
Algorithm 2 Kernelized Batch Mode Active Learning ($\mathrm{KLR_{BMAL}}$)
INPUT: L, U /* training data set, unlabeled data set */
        n, m, k /* unlabeled size, training size, batch size */
VARIABLES: $\mathbf{q}$, $I_p$ /* sampling probability, Fisher information matrix*/
PARAMETERS: $\delta$, $s$ /* regularization factor, number of top eigens */
OUTPUT: S /* a batch of selected unlabeled examples */
PROCEDURE:
initialize: S = $\emptyset$;   $q_i \leftarrow 1/n, i = 1,2,\ldots,n$;
$\theta$ = KLR_train(L);
$\pi_i = p(+|\mathbf{x}_i) = 1/(1+\exp(-\sum_{\mathbf{x}' \in \mathsf{L}} \theta(\mathbf{x}')K(\mathbf{x}',\mathbf{x}_i)))$
$\mathbf{g}_i = (K(\mathbf{x}_1^L,\mathbf{x}_i), K(\mathbf{x}_2^L,\mathbf{x}_i), \ldots, K(\mathbf{x}_m^L,\mathbf{x}_i)), i = 1,2,\ldots,n$
$I_p \leftarrow \frac{1}{n}\sum_{i=1}^{n} \pi_i(1-\pi_i)\mathbf{g}_i\mathbf{g}_i^\top + \delta I_m$;
($\mathbf{v}$, $\lambda$) = eig($I_p$, s); /* do eigen decomposition */
while (change in $\{q_i\} > \epsilon$) do
    $f_k \leftarrow \sum_{i=1}^{n} q_i\pi_i(1-\pi_i)(\mathbf{g}_i^\top \mathbf{v}_k)^2, k = 1,2,\ldots,s$;
    $q_i \leftarrow q_i^2\pi_i(1-\pi_i)\sum_{k=1}^{s}(\lambda_k \mathbf{g}_i^\top \mathbf{v}_k)^2/f_k^2$
    $q_i \leftarrow q_i/\sum_{j=1}^{n} q_j, i = 1,2,\ldots,n$;
end while
while (|S| $>=$ $k$) do
    $\mathbf{x}^* = \arg\max_{\mathbf{x}_i \in \mathsf{U}} q(\mathbf{x}_i)$;
    S $\leftarrow$ S $\cup$ $\{\mathbf{x}^*\}$;
    U $\leftarrow$ U - $\{\mathbf{x}^*\}$;
end while
return S;
```

Fig. 2. A kernelization algorithm for batch mode active learning

V. Batch Mode Active Learning for Text Categorization

In this section, we present an empirical study by applying the batch mode active learning technique with the bound optimization algorithm to text categorization applications.

A. Experimental Testbeds

Three text collections are used for this empirical study. For all three datasets, we remove both the stopwords and the numbers from the documents, and covert all the words into the lower case without stemming.

The first dataset is the Reuters-21578 Corpus, and more specifically, the ModApte split of the Reuters-21578. This text collection has been widely used for evaluating text categorization algorithms [39]. There are a total of 10788 text documents in this collection, and Table I shows a list of the 10 most frequent categories of this dataset. Since each document in the dataset can be assigned to multiple categories, we divide the multi-label text categorization problem into a number of binary classification problems, i.e., a different binary classification problem for each category. In total, $26,299$ word features are extracted and used to represent the text documents.

Category	earn	acq	money-fx	grain	crude
Size	485	478	283	286	237
Category	trade	interest	wheat	ship	corn
Size	3964	2369	717	582	578

TABLE I

A list of 10 major categories of the Reuters-21578 dataset.

Category	course	department	faculty	project	staff	student
Size	930	182	1124	504	137	1641

TABLE II

A list of 6 categories of the WebKB dataset.

Category	Cat-0	Cat-1	Cat-2	Cat-3	Cat-4	Cat-5
Size	1000	1000	1000	1000	1000	1000
Category	Cat-6	Cat-7	Cat-8	Cat-9	Cat-10	
Size	999	1000	1000	1000	997	

TABLE III

A list of 11 categories of the Newsgroup dataset.

The other two datasets are Web-related text collections: the WebKB dataset and the Newsgroup dataset. The WebKB dataset comprises of WWW-pages collected from computer science departments of various universities in January 1997 by the World Wide Knowledge Base (Web->Kb) project of the CMU text learning group. All the Web pages are classified into seven categories: student, faculty, staff, department, course, project, and others. In this study, we ignore the category "others" due to its unclear definition. In total, there are $4,518$ data samples in the selected WebKB dataset, and $19,686$ word features are extracted to represent the text documents. Table II shows the details of the WebKB dataset. The newsgroup dataset includes $20,000$ messages from 20 different newsgroups. Each newsgroup contains roughly 1000 messages. In this study, we randomly select 11 out of 20 newsgroups for evaluation. In total, there are $10,996$ data samples in the selected Newsgroup dataset, and $47,410$ word features are extracted to represent the text documents. Table III shows the details of the engaged dataset.

Compared to the Reuters-21578 dataset, the two Web-related data collections are different in that more unique words are found in the Web-related datasets. For example, both the Reuters-21578 dataset and the Newsgroup dataset, contain roughly $10,000$ documents. But, the number of unique words for the Newsgroups dataset is close to $50,000$, which is about twice as the number of unique words found in the Reuters-21578. Thus, it is more challenging to classify WWW documents than normal text documents because more feature weights need to be decided for the WWW documents. This feature also makes the active learning algorithms more valuable for classifying WWW documents than normal documents. This is because, by selecting informative documents for manual labeling, we can decide the appropriate weights for more words than by randomly selecting documents.

B. Experimental Settings and Compared Schemes

In order to remove the uninformative word features, feature selection is conducted using the Information Gain criterion [40]. In particular, top 500 most informative features are selected for each category in each of the three text collections described above.

For performance evaluation, the $F1$ metric is adopted as our evaluation metric, as it has proved to be a more reliable metric than other metrics such as the classification accuracy [40]. More specifically, the $F1$ is defined as $F1 = \frac{2*p*r}{p+r}$, where p and r are precision and recall, respectively. Note that the $F1$ metric takes into account both the precision and the recall, thus is a more comprehensive metric than either the precision or the recall separately.

To examine the effectiveness of the proposed active learning algorithm, we compare our algorithm with several existing algorithms, including two baseline random sampling methods, two typical active learning methods, and two online learning methods. First of all, both the logistic regression and the support vector machine models ($\mathbf{LR_{Rand}}$ and $\mathbf{SVM_{Rand}}$), trained on the initially labeled examples and randomly selected examples, are engaged in our experiments as the baseline models. By comparing to these two baseline approaches, we are able to determine the amount of benefit brought by different active learning algorithms.

Second, two popular active learning methods are studied, which are based on the LR and SVM models, respectively. One is the active learning algorithm based on the linear logistic regression model. It measures the classification uncertainty based on the entropy of the posterior distribution $p(y|\mathbf{x})$. In particular, for a given test example $\mathbf{x}$ and a logistic regression model, the entropy of the distribution $p(y|\mathbf{x})$ is calculated as:

$$H(p) = -p(-|\mathbf{x}) \log p(-|\mathbf{x}) - p(+|\mathbf{x}) \log p(+|\mathbf{x})$$

The larger the entropy of $\mathbf{x}$ is, the more uncertain we are about the class label of $\mathbf{x}$. We refer to this baseline model as the logistic regression active learning, or $\mathbf{LR_{AL}}$ for short. The second active learning model is based on the support vector machine [35] that has already been discussed in Section II. In this method, the classification uncertainty of an example $\mathbf{x}$ is determined by its distance to the decision boundary $\mathbf{w}^\top \mathbf{x} + b = 0$, i.e., $d(\mathbf{x}; \mathbf{w}, b) = \frac{|\mathbf{w}^\top \mathbf{x}+b|}{\|\mathbf{w}\|_2}$. The smaller the distance $d(\mathbf{x}; \mathbf{w}, b)$ is, the more the classification uncertainty will be. We refer to this approach as support vector machine active learning, or $\mathbf{SVM_{AL}}$ for short. In addition to the above two active learning algorithms, we also study their online version active learning algorithms, in which the classification model will be re-trained after every new example is actively selected. We denote these two online version algorithms as $\mathbf{LR_{Online}}$ and $\mathbf{SVM_{Online}}$, respectively.

To evaluate the performance of the compared active learning algorithms, we first randomly select 100 labeled documents, 50 positive examples and 50 negative examples, for each category from the dataset. Both the LR and the SVM models are trained on the 100 labeled documents initially. Each active learning method (except the two online algorithms) will select k additional documents for labeling, and is evaluated based on the classification model that is built upon a total of $100 + k$ labeled documents. For the two online algorithms, they select only 1 example in each iteration and repeat the selection for k iterations. Each experiment is carried out 40 times, and the averaged $F1$ together with its variance are calculated and used for final evaluation.

To deploy efficient implementations of our scheme toward large-scale text categorization tasks, all the algorithms used in this study are programmed in the C language. The testing hardware environment is on a Linux workstation with 3.2GHz CPU and 2GB physical memory. We employ the tool for logistic regression that are developed by Komarek and Moore [17]. To implement our active learning algorithm based on the bound optimization approach, we employ a standard math package for linear algebra, i.e., LAPACK [1], to solve the eigen decomposition. The SVM^{light} package [12] is used in our experiments for the implementation of SVM, which has been considered as one of the state-of-the-art tools for text categorization. Since SVM is not parameter-free and can be sensitive to the capacity parameter, a separate validation set is used to determine the optimal parameters for configuration.

C. Empirical Evaluation

In this subsection, we will first describe the results for the Reuters-21578 dataset, since this dataset has been most extensively studied for text categorization. We will then provide the empirical results for the two Web-related datasets.

1) Experimental Results with Reuters-21578: Table IV shows the experimental results of $F1$ performance averaging over 40 executions on 10 major categories in the dataset, in which each execution is given with 100 initial training samples and 10 additional samples by active learning.

First, as listed in the 1st and the 2nd columns of Table IV, we observe that the performance of the two baseline methods, $\mathbf{LR_{Rand}}$ and $\mathbf{SVM_{Rand}}$, are comparable when the two classifiers are trained with the same initially labeled 100 examples and an additional set of 10 randomly selected examples. For several categories, such as “grain”, “ship” and “corn”, $\mathbf{SVM_{Rand}}$ is considerably better than $\mathbf{LR_{Rand}}$.

Second, we compare the performance of the two regular active learning algorithms listed in the 3rd and 4th columns of Table IV, i.e., $\mathbf{LR_{AL}}$ and $\mathbf{SVM_{AL}}$, that use the margin as the selection criterion to select a batch of examples without retraining the classifiers. We found that the performance of these two active learning methods are rather close for most cases, except for a few categories, such as “money-fx”, “crude” and “ship”, where $\mathbf{LR_{AL}}$ performs better than $\mathbf{SVM_{AL}}$. By comparing them with the two random selection methods, both of them importantly outperform the random approaches.

Further, we examine the performance of the two online version algorithms $\mathbf{LR_{Online}}$ and $\mathbf{SVM_{Online}}$ that will retrain the classification model after an example is selected for labeling. We can see that their performances are close for most cases, except for a few categories, such as “money-fx”, crude” and “wheat”, where $\mathbf{LR_{Online}}$ considerably outperforms $\mathbf{SVM_{Online}}$. By comparing them with the two regular active learning algorithms, some interesting result was observed. As we know, it is commonly expected that an online algorithm usually is able to outperform the corresponding non-online approach importantly. But the two online algorithms only achieve relatively small improvements on this dataset. For some category, such as “acq”, the online algorithms are even slightly worse than the regular active learning solutions.

Finally, we evaluate the performance of the proposed algorithm $\mathbf{LR_{BMAL}}$, as shown in the last column of Table IV. We can see that $\mathbf{LR_{BMAL}}$ almost achieves the best results among the compared algorithms, except for the “earn” category, where the online algorithm $\mathbf{LR_{Online}}$ obtains the best result. The improvements by $\mathbf{LR_{BMAL}}$ are statistically significant over a number of categories, such as “trade”, “interest” and “corn”, according to the student’s t-test ($p < 0.05$). For the exceptional case, i.e., the “earn” category, $\mathbf{LR_{BMAL}}$ is slightly worse than the two online algorithms.

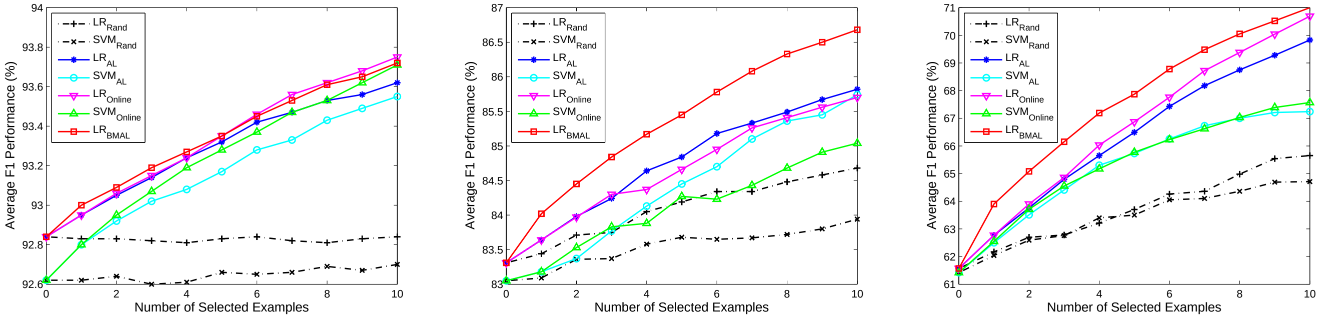
In order to examine the performance in more detail, we evaluate the $F1$ measure of each category by varying the number of actively selected examples from 1 to 10 for each of the compared algorithms. Fig. 3 and Fig. 4 show the experimental results of the $F1$ measurement. Similar observations can be drawn. First of all, we can see that all of the active learning algorithms significantly outperform the random selection algorithms. Second, among the compared active learning algorithms, the online algorithms outperform the other two regular algorithms on a number of categories, but the improvements are not always significant. The improvements of the online algorithms usually become more evident when the number of selected examples increases. This is consistent to the intuition that an online algorithm usually can perform better than a non-online approach since the margin criterion becomes more accurate after the classification model is retrained. Finally, by comparing the proposed algorithm to other algorithms, we found that $\mathbf{LR_{BMAL}}$ is consistently better than the two regular margin-based active learning algorithms for most situations. Finally, we can also see that $\mathbf{LR_{BMAL}}$ performs better than the two online algorithms for a number of categories, such as “acq”, “money-fx” and “corn”, etc.

2) Experimental Results with Web-Related Datasets: The classification results of the WebKB dataset and the Newsgroup dataset are listed in Tables V and VI. First, notice that for the two Web-related datasets, there are a few categories whose

Category	$\mathbf{LR_{Rand}}$	$\mathbf{SVM_{Rand}}$	$\mathbf{LR_{AL}}$	$\mathbf{SVM_{AL}}$	$\mathbf{LR_{Online}}$	$\mathbf{SVM_{Online}}$	$\mathbf{LR_{BMAL}}$
earn	92.84 ±0.12	92.70 ±0.11	93.62 ±0.09	93.55 ±0.10	**93.75** ±0.08	93.71 ±0.08	93.72 ±0.09
acq	84.68 ±0.21	83.94 ±0.21	85.82 ±0.23	85.74 ±0.18	85.70 ±0.18	85.04 ±0.27	**86.68** ±0.15
money-fx	65.65 ±0.55	64.71 ±0.49	69.83 ±0.45	67.24 ±0.43	70.69 ±0.41	67.57 ±0.48	**71.00** ±0.53
grain	62.72 ±0.89	65.76 ±0.68	69.22 ±0.69	69.09 ±0.47	69.99 ±0.64	69.62 ±0.45	**70.57** ±0.62
crude	70.53 ±0.34	69.55 ±0.34	74.17 ±0.43	71.09 ±0.45	74.07 ±0.33	71.90 ±0.31	**74.94** ±0.43
trade	51.37 ±0.64	52.87 ±0.43	55.44 ±0.62	55.49 ±0.41	55.90 ±0.53	55.92 ±0.41	**57.78** ±0.53
interest	57.63 ±0.59	58.95 ±0.59	61.21 ±0.52	61.96 ±0.48	62.05 ±0.43	62.00 ±0.41	**63.20** ±0.63
wheat	63.04 ±0.59	65.92 ±0.56	73.45 ±0.51	72.33 ±0.47	75.32 ±0.60	72.45 ±0.50	**75.76** ±0.52
ship	66.75 ±1.01	69.89 ±0.53	73.40 ±0.42	71.70 ±0.41	73.73 ±0.42	72.76 ±0.32	**74.14** ±0.36
corn	44.97 ±0.58	47.21 ±0.64	53.92 ±0.65	54.40 ±0.56	55.89 ±0.61	54.67 ±0.66	**58.14** ±0.99

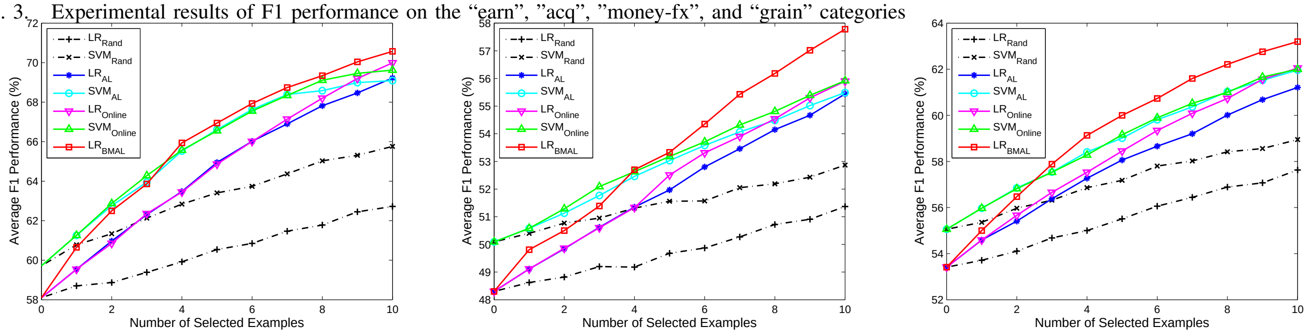
TABLE IV

THE F1 PERFORMANCE ON THE REUTERS-21578 DATASET WITH 100 TRAINING SAMPLES AND 10 ACTIVE SAMPLES(%).



(a) “earn” (b) “acq” (c) “money-fx”

Fig. 3. Experimental results of F1 performance on the “earn”, ”acq”, ”money-fx”, and “grain” categories



(a) “grain” (b) “trade” (c) “interest”

Fig. 4. Experimental results of F1 performance on the “trade”, “interest”, “corn” and “wheat” categories

Category	$\mathbf{LR_{Rand}}$	$\mathbf{SVM_{Rand}}$	$\mathbf{LR_{AL}}$	$\mathbf{SVM_{AL}}$	$\mathbf{LR_{Online}}$	$\mathbf{SVM_{Online}}$	$\mathbf{LR_{BMAL}}$
course	92.29 ±0.19	91.50 ±0.20	93.11 ±0.17	92.13 ±0.17	**93.33** ±0.15	92.34 ±0.21	93.24 ±0.17
department	78.95 ±0.65	78.98 ±0.55	85.19 ±0.43	84.86 ±0.28	85.69 ±0.38	85.36 ±0.37	**85.89** ±0.43
faculty	79.87 ±0.30	79.63 ±0.34	81.22 ±0.30	80.81 ±0.32	81.15 ±0.34	81.08 ±0.33	**81.59** ±0.31
project	68.53 ±0.38	67.19 ±0.64	70.51 ±0.38	69.78 ±0.50	70.76 ±0.42	70.25 ±0.51	**71.06** ±0.36
staff	23.40 ±0.42	22.42 ±0.50	25.56 ±0.47	25.56 ±0.64	26.22 ±0.47	**26.38** ±0.58	26.21 ±0.45
student	83.45 ±0.28	81.90 ±0.36	84.78 ±0.26	84.02 ±0.26	**84.92** ±0.27	83.80 ±0.29	84.88 ±0.26

TABLE V

THE F1 PERFORMANCE ON THE WEBKB DATASET WITH 100 TRAINING SAMPLES AND AND 10 ACTIVE SAMPLES (%).

$F1$ measurements are extremely low. For example, for the category “staff” of the WebKB dataset, the $F1$ measurement is only about 23% to 26% for all methods. This fact indicates that it is more challenging to classify the WWW documents than normal documents. Second, we observe that the two baseline methods, $\mathbf{LR_{Rand}}$ and $\mathbf{SVM_{Rand}}$, perform similarly on both Web datasets, in which the $\mathbf{LR_{Rand}}$ method slightly outperforms the $\mathbf{SVM_{Rand}}$ method for a few categories. Third, by comparing the two regular margin-based active learning approaches, namely, $\mathbf{LR_{AL}}$ and $\mathbf{SVM_{AL}}$, we observe that, for a number of categories, $\mathbf{LR_{AL}}$ achieves substantially better performance than $\mathbf{SVM_{AL}}$. The most noticeable case is the

Category	$\mathbf{LR_{Rand}}$	$\mathbf{SVM_{Rand}}$	$\mathbf{LR_{AL}}$	$\mathbf{SVM_{AL}}$	$\mathbf{LR_{Online}}$	$\mathbf{SVM_{Online}}$	$\mathbf{LR_{BMAL}}$
Cat-0	97.98 ±0.10	98.87 ±0.05	99.32 ±0.05	99.59 ±0.03	99.49 ±0.03	**99.72** ±0.02	99.40 ±0.05
Cat-1	92.64 ±0.25	92.09 ±0.38	95.18 ±0.14	94.77 ±0.16	95.29 ±0.11	94.81 ±0.16	**95.31** ±0.13
Cat-2	78.01 ±0.58	74.86 ±0.41	82.45 ±0.34	76.61 ±0.55	82.56 ±0.41	79.39 ±0.38	**82.71** ±0.33
Cat-3	86.67 ±0.55	84.57 ±0.40	90.55 ±0.29	87.33 ±0.34	**91.29** ±0.28	88.80 ±0.27	90.68 ±0.28
Cat-4	67.20 ±0.61	64.31 ±0.46	70.58 ±0.72	66.09 ±0.55	70.11 ±0.65	68.82 ±0.47	**71.34** ±0.71
Cat-5	76.43 ±0.22	73.84 ±0.29	77.87 ±0.20	76.16 ±0.25	77.88 ±0.25	76.39 ±0.27	**78.05** ±0.19
Cat-6	91.05 ±0.23	87.86 ±0.25	93.35 ±0.14	91.30 ±0.20	93.53 ±0.14	91.22 ±0.20	**93.61** ±0.13
Cat-7	59.10 ±0.93	55.66 ±1.09	63.72 ±0.89	61.23 ±0.99	64.97 ±0.76	61.30 ±1.05	**65.14** ±0.87
Cat-8	75.32 ±0.80	70.74 ±0.62	79.09 ±0.62	75.12 ±0.71	**79.83** ±0.45	77.94 ±0.45	79.27 ±0.59
Cat-9	79.97 ±0.50	78.28 ±0.40	83.71 ±0.41	82.04 ±0.46	**84.35** ±0.34	82.79 ±0.37	84.23 ±0.39
Cat-10	99.65 ±0.02	99.86 ±0.01	99.90 ±0.01	**99.95** ±0.00	99.94 ±0.00	**99.95** ±0.00	99.90 ±0.01

TABLE VI

THE F1 PERFORMANCE ON THE NEWSGROUP DATASET WITH 100 TRAINING SAMPLES AND 10 ACTIVE SAMPLES (%).

Dataset	$\mathrm{LR_{AL}}$	$\mathrm{SVM_{AL}}$	$\mathrm{LR_{Online}}$	$\mathrm{SVM_{Online}}$	$\mathrm{LR_{BMAL}}$
Reuters	0.237 ±0.005	0.328 ±0.010	2.481 ±0.005	3.364 ±0.016	0.882 ±0.009
WebKB	0.284 ±0.011	0.462 ±0.014	2.879 ±0.008	4.692 ±0.014	1.025 ±0.015
Newsgroup	0.530 ±0.011	0.872 ±0.020	5.496 ±0.011	8.791 ±0.019	1.571 ±0.015

TABLE VII

TIME PERFORMANCE ON THE THREE TEXT DATASETS (SECONDS).

category 2 of the Newsgroup datasets, where $\mathbf{SVM_{AL}}$ only a small improvement with the additional labeled examples. In contrast, the $\mathbf{LR_{AL}}$ algorithm improves the $F1$ measurement from 78.01% to 82.45%.

Finally, compared to $\mathbf{LR_{AL}}$, we observe that the proposed algorithm $\mathbf{LR_{BMAL}}$ is able to improve the $F1$ measurement considerably over the margin-based active learning approach in most cases. For example, for category 7 of the Newsgroup dataset, the $\mathbf{LR_{AL}}$ algorithm improves the baseline method from 59.10% to 63.72%, while the $\mathbf{LR_{BMAL}}$ algorithm is able to achieve a better improvement from 59.10% to 65.14%. Compared to the two online algorithms $\mathbf{LR_{Online}}$ and $\mathbf{SVM_{Online}}$, the proposed $\mathbf{LR_{BMAL}}$ algorithm perform closely to these two approaches on both Web datasets. For a number of categories, $\mathbf{LR_{BMAL}}$ performs better than the two online algorithms. This observation indicates that the proposed batch mode active learning algorithm is effective for large-scale text categorization tasks. It is important to note that this is not to claim the batch mode active learning algorithm is always better than the online algorithms. In fact, there are a few categories, the online algorithms are better than the proposed batch mode active learning method.

3) Time Performance: To further examine the efficiency of the proposed algorithm, we conduct experiments to evaluate the time performance compared with other active learning approaches. For each dataset, all algorithms are evaluated with an experiment of selecting 10 examples for active learning. Every experiment is repeated 40 executions. Table VII shows the experimental results of average time performance on the three text datasets. From the results, we observe that among the compared algorithms, the two regular active learning algorithms are the most efficient ones as they do not require additional retraining cost. In contrast, the two online algorithms are the least efficient ones, which dramatically increase the computational time. The proposed batch mode active learning algorithm, without retraining cost, achieves much smaller time cost compared to other two online algorithms (about 1/3∼1/4 fraction), though it is worse than the two regular active learning approaches (about 3∼4 times). This observation shows that the proposed algorithm is efficient for practical applications.

VI. BATCH MODE ACTIVE LEARNING FOR CBIR

In this section, we will apply the algorithm of batch mode active learning to relevance feedback in content-based image retrieval. As indicated in the related work, relevance feedback is critical to alleviating the semantic gap issue in CBIR, in which active learning has been shown to be one promising solution [34]. We will compare the proposed algorithm for batch mode active learning to the heuristic active learning methods for relevance feedback [34].

A. Experimental Testbed

To conduct empirical evaluation of our proposed algorithm, we choose the real-world images from the COREL image CDs. In total, we use $5,000$ images to form our testbed from 50 different image categories. Each category in the dataset consists of exactly 100 images that are randomly selected from the relevant examples in the COREL image CDs. Every category represents a different semantic topic, such as *dog*, *cat*, *horse*, *botany*, and *butterfly*, etc.

B. Image Representation

An important step of CBIR is the low-level feature extraction. Three kinds of features are extracted to represent the images in our experiments: color, edge and texture.

For color features, we use the color moment since it is close to natural human perception. Many previous research studies have shown the effectiveness of color moment applied in CBIR. Given an image, we extract 3 moments: color mean, color variance and color skewness in each color channel (H, S, and V), respectively. Thus, a 9-dimensional color moment is adopted as the color feature in our experiments.

For edge features, we employ the edge direction histogram. A given image is first converted to the gray image. Then a

Canny edge detector is applied to obtain the edge images, in which the edge direction histogram can be computed. The edge direction histogram is quantized into 18 bins of 20 degrees each; hence an 18-dimensional edge direction histogram is employed to represent the edge features.

For texture features, we study wavelet based textures. An input color image is first converted to the gray image. Then Discrete Wavelet Transformation (DWT) is performed on the gray image using a Daubechies-4 wavelet filter. Each wavelet decomposition on a gray 2D-image results in four subimages with a $0.5 * 0.5$ scaled-down image of the input image and the wavelets in three orientations. The scaled-down image is fed into the DWT operation to produce the next four subimages. In total, we perform 3-level decomposition and extract features from the 9 of the subimages by computing the entropy. Hence, a 9-dimensional wavelet texture is employed.

In total, a 36-dimensional feature vector is used to represent an image [9], including 9-dimension color features, 18-dimension edge features, and 9-dimension wavelet features.

C. Experimental Setup

In our experiments for CBIR, we have developed an algorithm of batch mode active learning, based on the kernel logistic regression classification model, to accomplish the relevance feedback function in our CBIR systems. For simplicity, we will refer to our batch mode active learning algorithm as $\mathbf{KLR_{BMAL}}$. To evaluate the effectiveness of our batch mode active learning algorithm for relevance feedback of CBIR, similar to the previous experiment, we compare our algorithm to two random selection algorithms $\mathbf{KLR_{Rand}}$ and $\mathbf{SVM_{Rand}}$, and two regular active learning algorithms $\mathbf{SVM_{AL}}$ and $\mathbf{KLR_{AL}}$. Similarly, two online version algorithms $\mathbf{KLR_{Online}}$ and $\mathbf{SVM_{Online}}$ are also implemented, although they are seldom used for relevance feedback in CBIR. For all the classification models used in this study, the same RBF kernel is used. The kernel width is determined by separate validation sets.

In our experiments, relevance judgments are based on their semantic categories of images. In particular, an image is judged as relevant to a query when both the image and the query example belong to the same semantic category. Although this definition is somewhat simplified, it does allow us to evaluate the retrieval performance automatically and systematically, thus reducing subjective errors arising from manual evaluations by different people. The similar approach has been widely adopted by previous studies [34], [9].

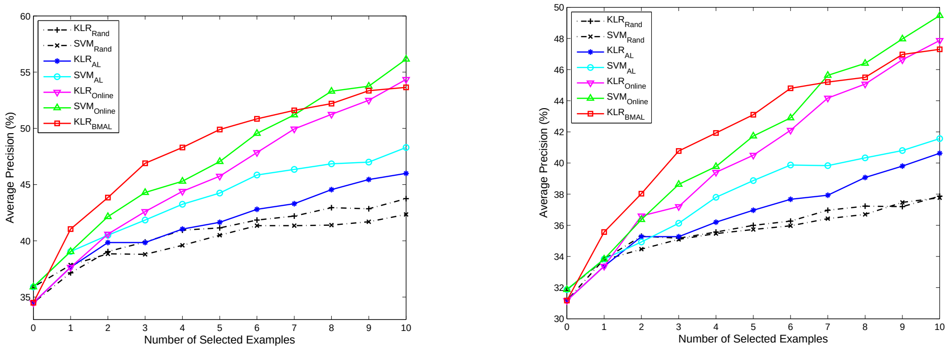
To enable objective comparisons, we simulate the relevance feedback of CBIR as follows. We first randomly select a query image from the Corel database, and retrieve the N_{init} images that are closest to the query example in terms of Euclidean distance. We then simulate the user's relevance feedback for the N_{init} retrieved images based on their semantic categories. The retrieved images are marked as relevant when both the retrieved image and the query examples share the same categories. Next, we apply the active learning algorithms to identify the N_{batch} (the batch size) most informative images for manual labeling in each iteration of active learning. A classification model is built based on both the initially labeled images and the labeled images that are acquired by active learning. The final retrieval results are ranked based on the learned classification model. The *Average Precision* is used as the evaluation metric in our experiments. This is defined as the percentage of returned images that are relevant to the query examples.

D. Performance Evaluation

In our experiments, we evaluate the active learning algorithms with respect to the change of the number of selected examples, i.e., the batch size N_{batch}. We randomly pick 100 image examples from the testbed as the queries. For each query, the number of initially labeled images N_{init} is set to 10 and the experimental results of relevance feedback using different active learning algorithms are evaluated by changing the number of selected examples N_{batch} from 1 to 10. Fig 5 (a) and (b) show the experimental results of average precision on both top 20 and top 30 ranked results respectively.

Several observations can be drawn from the experimental results in Fig 5. First of all, comparing the two random selection algorithms, we found that their retrieval results are similar no matter how the batch size changes. For the top 20 ranked results, $\mathbf{KLR_{Rand}}$ is slightly better than $\mathbf{SVM_{Rand}}$, but their difference becomes smaller for the top 30 ranked results. Second, for the two regular margin-based active learning algorithms $\mathbf{KLR_{AL}}$ and $\mathbf{SVM_{AL}}$, both of them considerably outperform the random approaches particularly when N_{batch} increases. In most cases, especially when N_{batch} is larger than 3, $\mathbf{SVM_{AL}}$ is better than $\mathbf{KLR_{AL}}$ on both the top 20 and top 30 ranked results. Further, for the two online algorithms $\mathbf{KLR_{Online}}$ and $\mathbf{SVM_{Online}}$, both of them are consistently better than their non-online algorithms. The improvements become more evident when N_{batch} increases. This observation matches our intuition that the online algorithms often work better given more accurate classification models by re-training. Finally, comparing the proposed algorithm $\mathbf{KLR_{BMAL}}$ to other algorithms, we can see that $\mathbf{KLR_{BMAL}}$ achieves the best results when the number of selected examples is smaller than 6. For all cases, $\mathbf{KLR_{BMAL}}$ is significantly better than the regular active learning algorithm $\mathbf{KLR_{AL}}$. Compared to the online algorithms $\mathbf{KLR_{Online}}$ and $\mathbf{SVM_{Online}}$, $\mathbf{KLR_{BMAL}}$ considerably outperform the two online algorithms when N_{batch} is smaller than 6. When N_{batch} is greater than 6, the improvement become smaller. When N_{batch} is set to 10, $\mathbf{KLR_{BMAL}}$ fails to improve over the two online algorithms.

To examine the effectiveness of the proposed algorithm in more details, we also evaluate the average precision of other top ranked results. Table VIII and Table IX show the experimental results of average precision on top 20 $\sim$ 100 with $N_{batch} = 5$ and $N_{batch} = 10$, respectively. For the results with $N_{batch} = 5$, the proposed algorithm $\mathbf{KLR_{BMAL}}$ achieves the best performance from top 20 to top 50 ranked results, which are usually more critical for a CBIR task. The improvements on these results are statistically significant according to the student's t-test ($p < 0.05$). For the results with $N_{batch} = 10$, $\mathbf{KLR_{BMAL}}$ outperforms the two regular



(a) Top 20 ranked results

(b) Top 30 ranked results

Fig. 5. Experimental results of several active learning algorithms for content-based image retrieval.

active learning algorithms consistently, but does not achieve any improvement over the two online algorithms.

Remark. As an important note, we need to mention that the main purpose of engaging the two online algorithms for comparisons is to let us judge how good is the performance achieved by a batch mode active learning algorithm. We emphasize that the BMAL algorithm enjoys two major advantages over the online algorithms. First, no retraining cost is required for the batch mode active learning algorithm. Second, the BMAL algorithm avoids the additional overhead of user interventions as required by an online algorithm, which is inefficient and not practical for real-world applications. This is because every relevance feedback procedure between a system and a user often has to incur some overhead in either system response or network communication time (network overhead for a Web application). This is why most practical relevance feedback solutions in CBIR are usually conducted in a batch fashion but not an online fashion [34], [9].

VII. Comparison of Two Optimization Methods

In the two preceding subsections, we have shown that the algorithm of batch mode active learning using bound optimization is effective for both text categorization and relevance feedback of CBIR. In this section, we conduct an empirical study of two different implementations of batch mode active learning, for which both of them are based on the kernel logistic regression model. One is the approximated bound optimization approach ($\mathrm{KLR}^{\mathrm{BO}}_{\mathrm{BMAL}}$), and the other is the original semi-definite programming (SDP) approach ($\mathrm{KLR}^{\mathrm{SDP}}_{\mathrm{BMAL}}$).

A. Experimental Testbed

Due to the high computational cost of SDP, we do not use text categorization and relevance feedback of CBIR for evaluation in this study. Instead, we choose 5 datasets of relatively small size from UCI machine learning repository as our experimental testbed. Table X shows the datasets in our experiment.

Dataset	**#Classes**	**#Instances**	**#Features**
Breast-cancer	2	683	9
Cleveland	2	297	13
Heart	2	270	13
House-votes	2	435	16
Ionosphere	2	351	34

TABLE X
The UCI machine learning datasets in our testbed.

B. Experimental Results

The purpose of this experiment is to compare the effectiveness of two batch mode active learning (BMAL) implementations using different optimization formulations, i.e., ($\mathrm{KLR}^{\mathrm{BO}}_{\mathrm{BMAL}}$) and ($\mathrm{KLR}^{\mathrm{SDP}}_{\mathrm{BMAL}}$). For comparison, we also compare them with a KLR margin-based active learning solution $\mathbf{KLR_{AL}}$, which has been described in the previous section.

In our experimental settings, a set of initial 20 random training examples is provided for training a logistic regression classifier. After the initial classifier is obtained, active learning algorithms are engaged for selecting a batch of 20 informative examples for labeling. Finally, an updated logistic regression classifier is re-trained on the combined set of the initial training examples and the additional batch of examples.

Datasets	Metrics	$\mathrm{KLR}_{\mathrm{AL}}$	$\mathrm{KLR}^{\mathrm{SDP}}_{\mathrm{BMAL}}$	$\mathrm{KLR}^{\mathrm{BO}}_{\mathrm{BMAL}}$
Cleveland	Acc.	76.70 ± 0.55	**78.74 ± 0.70**	78.62 ± 0.54
	F1	75.19 ± 0.60	76.33 ± 0.71	76.20 ± 0.57
Heart	Acc.	76.95 ± 0.64	**79.73 ± 0.46**	79.55 ± 0.36
	F1	74.24 ± 0.74	76.20 ± 0.59	**76.24 ± 0.45**
House	Acc.	93.18 ± 0.42	**94.48 ± 0.39**	94.01 ± 0.33
	F1	94.43 ± 0.35	**95.39 ± 0.33**	95.00 ± 0.28
Ionosphere	Acc.	76.52 ± 0.54	77.77 ± 0.50	**77.87 ± 0.65**
	F1	83.70 ± 0.39	84.34 ± 0.37	**84.80 ± 0.52**

TABLE XI
Experimental results of classification performance evaluation on UCI testbeds (%).

Table XI shows the experimental results of the performance comparison. From the results, we found that both of two different BMAL implementations are considerably more effective than $\mathbf{KLR_{AL}}$, which again validated the similar results as achieved beforehand. By comparing the performance of the two BMAL implementations, we can see that the performance of the bound optimization approach ($\mathrm{KLR}^{\mathrm{BO}}_{\mathrm{BMAL}}$) is very similar to the original approach by SDP optimization ($\mathrm{KLR}^{\mathrm{BO}}_{\mathrm{BMAL}}$). It is somewhat surprising that the $\mathrm{KLR}^{\mathrm{BO}}_{\mathrm{BMAL}}$ solution is slightly better than the SDP solution in some cases, such as the "Heart" and "Ionosphere" datasets. These results show that the bound optimization solution is empirically a good approximation of the original formulation.

Finally, we evaluate the computational efficiency of the two different implementations of batch mode active learning. Both algorithms are run in Matlab environment with a PC of 3.2GHz CPU. In our implementation, the $\mathrm{KLR}^{\mathrm{SDP}}_{\mathrm{BMAL}}$ algo-

TOP	$\mathbf{KLR_{Rand}}$	$\mathbf{SVM_{Rand}}$	$\mathbf{KLR_{AL}}$	$\mathbf{SVM_{AL}}$	$\mathbf{KLR_{Online}}$	$\mathbf{SVM_{Online}}$	$\mathbf{KLR_{BMAL}}$
20	41.15 ±0.43	40.50 ±0.42	41.65 ±0.43	44.25 ±0.44	45.75 ±0.44	47.05 ±0.47	**49.90** ±0.47
30	36.00 ±0.37	35.73 ±0.38	36.97 ±0.39	38.87 ±0.39	40.50 ±0.41	41.73 ±0.42	**43.10** ±0.41
40	32.18 ±0.34	32.10 ±0.34	33.55 ±0.36	34.45 ±0.35	36.93 ±0.38	37.43 ±0.37	**38.28** ±0.37
50	29.58 ±0.30	29.66 ±0.31	30.38 ±0.32	31.42 ±0.32	33.28 ±0.33	33.78 ±0.33	**34.56** ±0.34
60	27.38 ±0.27	27.63 ±0.29	28.17 ±0.29	28.75 ±0.29	30.92 ±0.30	31.22 ±0.31	**31.43** ±0.31
70	25.60 ±0.25	26.10 ±0.26	26.29 ±0.27	26.39 ±0.27	28.83 ±0.28	**29.07** ±0.28	**29.07** ±0.28
80	24.23 ±0.24	24.88 ±0.24	24.99 ±0.25	24.85 ±0.25	27.13 ±0.26	**27.29** ±0.26	27.15 ±0.26
90	22.78 ±0.22	23.36 ±0.22	23.56 ±0.24	23.63 ±0.24	25.71 ±0.25	**25.90** ±0.25	25.78 ±0.25
100	21.72 ±0.21	22.19 ±0.21	22.26 ±0.22	22.31 ±0.22	24.43 ±0.23	**24.68** ±0.23	24.52 ±0.23

TABLE VIII

AVERAGE PRECISION OF THE COMPARED ALGORITHMS FOR IMAGE RETRIEVAL WITH 5 SELECTED EXAMPLES(%).

TOP	$\mathbf{KLR_{Rand}}$	$\mathbf{SVM_{Rand}}$	$\mathbf{KLR_{AL}}$	$\mathbf{SVM_{AL}}$	$\mathbf{KLR_{Online}}$	$\mathbf{SVM_{Online}}$	$\mathbf{KLR_{BMAL}}$
20	43.75 ±0.44	42.35 ±0.43	46.00 ±0.45	48.30 ±0.46	54.35 ±0.48	**56.15** ±0.51	53.65 ±0.47
30	37.87 ±0.39	37.77 ±0.39	40.63 ±0.42	41.57 ±0.41	47.87 ±0.46	**49.47** ±0.47	47.30 ±0.44
40	33.88 ±0.35	34.45 ±0.36	36.73 ±0.38	37.58 ±0.38	43.90 ±0.43	**45.25** ±0.43	41.10 ±0.40
50	31.20 ±0.31	31.78 ±0.33	33.22 ±0.35	34.16 ±0.35	39.68 ±0.40	**40.80** ±0.39	36.84 ±0.36
60	29.05 ±0.28	29.77 ±0.31	30.98 ±0.32	31.13 ±0.32	36.72 ±0.37	**37.42** ±0.37	33.62 ±0.32
70	27.10 ±0.26	27.94 ±0.28	29.09 ±0.30	28.94 ±0.29	34.03 ±0.34	**34.61** ±0.34	31.19 ±0.30
80	25.43 ±0.25	26.51 ±0.26	27.36 ±0.28	26.86 ±0.27	31.76 ±0.32	**32.29** ±0.32	29.34 ±0.28
90	24.17 ±0.23	25.14 ±0.24	25.71 ±0.26	25.49 ±0.26	29.91 ±0.30	**30.80** ±0.30	27.71 ±0.26
100	23.02 ±0.22	23.77 ±0.22	24.21 ±0.24	23.92 ±0.24	28.49 ±0.28	**28.99** ±0.28	26.22 ±0.25

TABLE IX

AVERAGE PRECISION OF THE COMPARED ALGORITHMS FOR IMAGE RETRIEVAL WITH 10 SELECTED EXAMPLES(%).

Algorithm	Cleveland	Heart	House-Votes	Ionosphere
$\mathrm{KLR}^{\mathrm{BO}}_{\mathrm{BMAL}}$	0.105	0.096	0.185	0.349
$\mathrm{KLR}^{\mathrm{SDP}}_{\mathrm{BMAL}}$	8.720	7.357	28.103	103.516

TABLE XII

COMPUTATIONAL TIME OF TWO BMAL FORMULATIONS (SECONDS).

rithm is implemented by the SeDuMi packages [32], a popular and efficient solution to SDP problems in Matlab. Table XII shows the computational time of the two algorithms averaged over 40 executions on each of the UCI datasets. It is clear that the $\mathrm{KLR}^{\mathrm{BO}}_{\mathrm{BMAL}}$ algorithm using the bound optimization formulation is significantly more efficient than $\mathrm{KLR}^{\mathrm{SDP}}_{\mathrm{BMAL}}$, i.e., the algorithm using SDP formulation. More specifically, the $\mathrm{KLR}^{\mathrm{BO}}_{\mathrm{BMAL}}$ algorithm is about $77 \sim 297$ times faster than the $\mathrm{KLR}^{\mathrm{SDP}}_{\mathrm{BMAL}}$ algorithm across different datasets.

VIII. CONCLUSIONS

This paper presents a framework of batch mode active learning for data classification and multimedia retrieval. Unlike the traditional active learning approach, which focuses on selecting a single example in each iteration, the batch mode active learning approach allows for multiple examples to be selected simultaneously for manual labeling. We employ the Fisher information matrix for the measurement of model uncertainty, and choose the set of examples that will effectively increase the Fisher information. To solve the related optimization problem effectively, we first formulate the learning problem into a semi-definite programming problem. We then develop an effective algorithm of batch mode active learning based on the bound optimization technique. Furthermore, we develop a kernel version of batch mode active learning using the kernel logistic regression. We apply our method to large-scale text categorization and relevance feedback of content-based image retrieval, and show promising results in comparison to two state-of-the-art active learning algorithms. Our empirical study also shows that the approximated algorithm of batch mode active learning using bound optimization performs as well as the SDP version of batch mode active learning. In future work, we will extend our methodology to other machine learning problems.

ACKNOWLEDGMENT

The authors would like to thank Dr. Paul Komarek in CMU for sharing part of the text datasets, and for his help in deploying the logistic regression package. The work was supported in part by the National Science Foundation (IIS-0643494), National Institute of Health (1R01-GM079688-01), and Singapore MOE Academic Tier-1 Research Grant (RG67/07). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF and NIH.

APPENDIX

APPENDIX A. PROOF OF INEQUATION

In this appendix we prove the following inequality from Section IV.F. Let $\mathcal{L}(\mathbf{q})$ be the objective function in (19), we then have

$$\begin{aligned}\mathcal{L}(\mathbf{q}) &= \sum_{k=1}^{s} \frac{\lambda_k}{\sum_{i=1}^{n} q_i \pi_i (1-\pi_i)(\mathbf{x}_i^\top \mathbf{v}_k)^2} \\ &= \sum_{k=1}^{s} \frac{\lambda_k}{\sum_{i=1}^{n} q'_i \pi_i (1-\pi_i)(\mathbf{x}_i^\top \mathbf{v}_k)^2} \times \frac{\sum_{i=1}^{n} q'_i \pi_i (1-\pi_i)(\mathbf{x}_i^\top \mathbf{v}_k)^2}{\sum_{i=1}^{n} q'_i \pi_i (1-\pi_i)(\mathbf{x}_i^\top \mathbf{v}_k)^2 \frac{q_i}{q'_i}}\end{aligned} \quad (25)$$

Using the convexity property of reciprocal function, namely

$$1/\sum_{i=1}^{n} p_i x \leq \sum_{i=1}^{n} \frac{p_i}{x} \qquad (26)$$

for $x \geq 0$ and pdf $\{p_i\}_{i=1}^n$, we can arrive at the following deduction:

$$\begin{aligned}\frac{\sum_{i=1}^n q'_i \pi_i (1-\pi_i)(\mathbf{x}_i^\top \mathbf{v}_k)^2}{\sum_{i=1}^n q'_i \pi_i (1-\pi_i)(\mathbf{x}_i^\top \mathbf{v}_k)^2 \frac{q_i}{q'_i}} & \leq \sum_{i=1}^n \frac{q'_i \pi_i (1-\pi_i)(\mathbf{x}_i^\top \mathbf{v}_k)^2}{\sum_{j=1}^n q'_j \pi_j (1-\pi_j)(\mathbf{x}_j^\top \mathbf{v}_k)^2} \frac{1}{\frac{q_i}{q'_i}} \\ & = \sum_{i=1}^n \frac{(q'_i)^2 \pi_i (1-\pi_i)(\mathbf{x}_i^\top \mathbf{v}_k)^2}{q_i \sum_{j=1}^n q'_j \pi_j (1-\pi_j)(\mathbf{x}_j^\top \mathbf{v}_k)^2}\end{aligned}$$

Substituting the above inequation back into (25), we can achieve the following inequality:

$$\begin{aligned} & \mathcal{L}(\mathbf{q}) \\ \leq & \sum_{k=1}^{s} \frac{\lambda_k}{\sum_{i=1}^n q'_i \pi_i (1-\pi_i)(\mathbf{x}_i^\top \mathbf{v}_k)^2} \times \left(\sum_{i=1}^n \frac{(q'_i)^2 \pi_i (1-\pi_i)(\mathbf{x}_i^\top \mathbf{v}_k)^2}{q_i \sum_{j=1}^n q'_j \pi_j (1-\pi_j)(\mathbf{x}_j^\top \mathbf{v}_k)^2} \right) \\ = & \sum_{k=1}^{s} \frac{\lambda_k}{\left(\sum_{j=1}^n q'_j \pi_j (1-\pi_j)(\mathbf{x}_j^\top \mathbf{v}_k)^2\right)^2} \times \sum_{i=1}^n \frac{(q'_i)^2 (\mathbf{x}_i^\top \mathbf{v}_k)^2 \pi_i (1-\pi_i)}{q_i} \\ = & \sum_{i=1}^{n} \frac{(q_i'^2)}{q_i} \pi_i (1-\pi_i) \sum_{k=1}^{s} \frac{(\mathbf{x}_i^\top \mathbf{v}_k)^2 \lambda_k}{(\sum_{j=1}^n q'_j \pi_j (1-\pi_j)(\mathbf{x}_j^\top \mathbf{v}_k)^2)^2} \ . \end{aligned}$$

This finishes the proof of the inequality mentioned above. ■

REFERENCES

[1] E. Z. B. Anderson. *LAPACK user's guide (3rd ed.)*. Philadelphia, PA, SIAM, 1999.

[2] C. Apte, F. Damerau, and S. Weiss. Automated learning of decision rulesfor text categorization. *ACM Trans. on Information Systems*, 12(3):233–251, 1994.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2003.

[4] Colin Campbell, Nello Cristianini, and Alex J. Smola. Query learning with large margin classifiers. In *17th International Conference on Machine Learning (ICML)*, pages 111–118, San Francisco, CA, 2000.

[5] William W. Cohen. Text categorization and relational learning. In *12th International Conference on Machine Learning (ICML)*, pages 124–132, 1995.

[6] Shai Fine, Ran Gilad-Bachrach, and Eli Shamir. Query by committee, linear separation and random walks. *Theor. Comput. Sci.*, 284(1):25–51, 2002.

[7] Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Mach. Learn.*, 28(2-3):133–168, 1997.

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[9] Steven C. H. Hoi, Michael R Lyu, and Rong Jin. A unified log-based relevance feedback scheme for image retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 18(4):509–524, 2006.

[10] Steven C.H. Hoi, Rong Jin, and Michael R. Lyu. Large-scale text categorization by batch mode active learning. In *Proc. the 15th International World Wide Web conference (WWW2006)*, Edinburgh, England, UK, May 23–26 2006.

[11] T. S. Huang and X. S. Zhou. Image retrieval by relevance feedback: from heuristic weight adjustment to optimal learning methods. In *Proc. IEEE Int. Conference on Image Processing*, volume 3, pages 2–5, October 2001.

[12] T. Joachims. Making large-scale svm learning practical. In *Advances in Kernel Methods - Support Vector Learning*, MIT Press, 1999.

[13] Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proc. 10th European Conference on Machine Learning (ECML)*, number 1398, pages 137–142, 1998.

[14] Michael I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Comput.*, 6(2):181–214, 1994.

[15] G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971.

[16] P. Komarek and A. Moore. Fast robust logistic regression for large sparse datasets with binary outputs. In *Artificial Intelligence and Statistics (AISTAT)*, 2003.

[17] Paul Komarek and Andrew Moore. Making logistic regression a core data mining tool: A practical investigation of accuracy, speed, and simplicity. In *Technical Report TR-05-27 at the Robotics Institute, Carnegie Mellon University*, May 2005.

[18] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proc.17th ACM International SIGIR Conference*, pages 3–12, 1994.

[19] R. Liere and P. Tadepalli. Active learning with committees for text categorization. In *Proceedings 14th Conference of the American Association for Artificial Intelligence (AAAI)*, pages 591–596, MIT Press, 1997.

[20] T. Luo, K. Kramer, S. Samson, and A. Remsen. Active learning to recognize multiple types of plankton. pages III: 478–481, 2004.

[21] S. MacArthur, C. Brodley, and C. Shyu. Relevance feedback decision trees in content-based image retrieval. In *Proc. IEEE Workshop on Content-based Access of lmage and Video Libraries*, pages 68–72, 2000.

[22] D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

[23] B. Masand, G. Lino, and D. Waltz. Classifying news stories using memory based reasoning. In *15th ACM SIGIR Conference*, pages 59–65, 1992.

[24] Andrew Kachites McCallum and Kamal Nigam. Employing EM and pool-based active learning for text classification. In *Proc.15th International Conference on Machine Learning*, pages 350–358. San Francisco, CA, 1998.

[25] Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *18th International Conference on Machine Learning (ICML)*, pages 441–448, 2001.

[26] Miguel E. Ruiz and Padmini Srinivasan. Hierarchical text categorization using neural networks. *Information Retrieval*, 5(1):87–118, 2002.

[27] Greg Schohn and David Cohn. Less is more: Active learning with support vector machines. In *Proc. 17th International Conference on Machine Learning*, pages 839–846, 2000.

[28] H. S. Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Computational Learning Theory*, pages 287–294, 1992.

[29] Xuehua Shen and ChengXiang Zhai. Active feedback in ad hoc information retrieval. In *Proc. ACM SIGIR'05*, pages 59–66, 2005.

[30] S. D. Silvey. *Statistical Inference*. 1974.

[31] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(12):1349–1380, 2000.

[32] J.F. Sturm. Using sedumi: a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[33] K. Tieu and P. Viola. Boosting image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, volume 1, pages 228–235, South Carolina, USA, 2000.

[34] Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM International Conference on Multimedia*, pages 107–118, Ottawa, Canada, 2001.

[35] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. In *Proc. 17th International Conference on Machine Learning (ICML)*, pages 999–1006, Stanford, US, 2000.

[36] Konstadinos Tzeras and Stephan Hartmann. Automatic indexing based on Bayesian inference networks. In *Proc. 16th ACM Int. SIGIR Conference*, pages 22–34, 1993.

[37] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[38] N. Vasconcelos and A. Lippman. Learning from user feedback in image retrieval systems. In *Advances in Neural Information Processing Systems*, 1999.

[39] Y. Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1(1/2):67–88, 1999.

[40] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proc. 14th International Conference on Machine Learning (ICML)*, pages 412–420, Nashville, 1997.

[41] Jian Zhang, Rong Jin, Yiming Yang, and Alex Hauptmann. Modified logistic regression: An approximation to svm and its applications in large-scale text categorization. In *International Conference on Machine Learning*, 2003.

[42] L. Zhang, F. Lin, and B. Zhang. Support vector machine learning for image retrieval. In *Proceedings of the International Conference on Image Processing (ICIP 2001)*, volume 2, pages 721–724, 2001.

[43] T. Zhang and F. J. Oles. A probability analysis on the value of unlabeled data for classification problems. In *17th International Conference on Machine Learning (ICML)*, 2000.

[44] J. Zhu. Semi-supervised learning literature survey. Technical report, Carnegie Mellon University, 2005.

[45] Ji Zhu and Trevor Hastie. Kernel logistic regression and the import vector machine. In *Advances in Neural Information Processing Systems 14*, pages 1081–1088, 2001.