Semi-Supervised SVM Batch Mode Active Learning with Applications to Image Retrieval

Steven C.H. Hoi
Nanyang Technological University
Rong Jin
Michigan State University
Jianke Zhu
Chinese University of Hong Kong
Michael R. Lyu
Chinese University of Hong Kong

Support vector machine (SVM) active learning is one popular and successful technique for relevance feedback in content-based image retrieval (CBIR). Despite the success, conventional SVM active learning has two main drawbacks. First, the performance of SVM is usually limited by the number of labeled examples. It often suffers a poor performance for the small-sized labeled examples, which is the case in relevance feedback. Second, conventional approaches do not take into account the redundancy among examples, and could select multiple examples that are similar (or even identical). In this work, we propose a novel scheme for explicitly addressing the drawbacks. It first learns a kernel function from a mixture of labeled and unlabeled data, and therefore alleviates the problem of small-sized training data. The kernel will then be used for a batch mode active learning method to identify the most informative and diverse examples via a min-max framework. Two novel algorithms are proposed to solve the related combinatorial optimization: the first approach approximates the problem into a quadratic program, and the second solves the combinatorial optimization approximately by a greedy algorithm that exploits the merits of submodular functions. Extensive experiments with image retrieval using both natural photo images and medical images show that the proposed algorithms are significantly more effective than the state-of-the-art approaches.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.4.0 [**Information Systems Applications**]: General; I.2.6 [**Computing Methodologies**]: Artificial Intelligence

General Terms: Algorithms, Design, Experimentation, Performance

Author's address: Steven C.H. Hoi, Block N4, Room #02c-112, Division of Information Systems, School of Computer Engineering, Nanyang Technological University, Singapore 639798
Rong Jin,Department of Computer Science and Engineering, 3115 Engineering Building, Michigan State University, East Lansing, MI 48824, U.S.A.
Jianke Zhu and Michael R. Lyu, Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong S.A.R.
A short version was appeared in the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR2008), 2008.

Additional Key Words and Phrases: content-based image retrieval, support vector machines, active learning, semi-supervised learning, batch mode active learning, human-computer interaction

1. INTRODUCTION

Relevance feedback [Rui et al. 1998] is the key technique that improves the accuracy of content-based image retrieval (CBIR) by exploiting the users' interaction with CBIR systems. In particular, users are encouraged to provide relevance judgments for the images retrieved by CBIR systems, and relevance feedback algorithms are designed to learn and understand users' information needs from the judged images [Smeulders et al. 2000; Lew et al. 2006]. One important research question related to relevance feedback is to decide which images should be presented to the users for maximizing the information gained. To this end, active learning has been proposed to identify the image examples that could be most helpful for understanding users' information needs. This is in contrast to passive relevance feedback, where only the images with the highest relevance scores are presented to users. A popular approach toward active relevance feedback in CBIR is support vector machine (SVM) active learning [Tong and Chang 2001]. This learns an SVM model from feedback examples, and employs the learned SVM model to identify the informative image examples for relevance feedback. Empirical studies showed that SVM active learning outperformed passive relevance feedback significantly in CBIR [Tong and Chang 2001; Panda et al. 2006; Rui et al. 1998].

Despite this success, conventional SVM active learning is limited by two major shortcomings when deployed for relevance feedback in CBIR. First, the performance of SVM is usually limited by the number of training data. When the number of labeled examples is small, which is the case in relevance feedback, conventional SVM may deliver poor classification accuracy, which could significantly affect the performance of SVM active learning. Second, in each round of relevance feedback, multiple image examples are presented to users for relevance judgments. Since conventional SVM active learning is designed to select a single example for each learning iteration, it may select similar images when applied to the task of choosing multiple examples. We refer to these two problems as the "*small training size problem*" and the "*batch sampling problem*", respectively.

To address the above problems, we propose a novel scheme for active learning, termed **Semi-Supervised Support Vector Machine Batch Mode Active Learning**. Our scheme handles the small training size problem by a semi-supervised learning technique, and the batch sampling problem in active learning by a min-max framework. In addition, we present two algorithms to efficiently solve the related combinatorial optimization problem, one by a quadratic programming technique and the other by submodular functions. Our extensive empirical study shows encouraging results in comparison to the state-of-the-art active learning algorithms for relevance feedback.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the problem formulation and our solution. Section 4 gives extensive evaluations in CBIR. Section 5 concludes this work.

2. RELATED WORK

Learning with relevance feedback in CBIR has been extensively studied, and has been shown as one way to attack the semantic gap issue [Smeulders et al. 2000; Rui et al. 1998]. From a general machine learning view, existing relevance feedback techniques can be grouped into two categories: *passive learning* versus *active learning*. In the past decade, a wide variety of techniques have been proposed for relevance feedback with passive learning approaches. Some earlier techniques include the well-known MARS [Rui et al. 1997], MindReader [Ishikawa et al. 1998], and some query re-weighting approaches [Rui et al. 1998], etc. Along with the prosperity of machine learning research in recent years, various passive machine learning methods have been applied to relevance feedback, including Bayesian learning [Vasconcelos and Lippman 1999], decision tree [MacArthur et al. 2000], boosting [Tieu and Viola 2000], discriminant analysis [Zhou and Huang 2001], incremental kernel biased discriminant analysis [Tao et al. 2006], negative samples analysis [Tao et al. 2007], nonparametric discriminant analysis [Tao and Tang 2004a], null-space analysis [Tao et al. 2008], Self-organizing map (SOM) [Laaksonen et al. 1999], EM algorithms [Wu et al. 2000], Gaussian mixture model [Qian et al. 2002], and Support Vector Machines (SVM) [Zhang et al. 2001; Hong et al. 2000; Tao and Tang 2004b; Hoi et al. 2006], among others. Because of limited space, we are unable to enumerate all existing approaches; more passive learning techniques for relevance feedback can be found in [Huang and Zhou 2001; Zhou and Huang 2003; Lew et al. 2006]. Among various solutions, the SVM based method might be one of the most active research topics for relevance feedback due to its solid theory [Vapnik 1998] and excellent generalization performance in real applications.

In contrast to the passive learning techniques, active learning has recently been actively studied with the aim of improving the learning efficiency of relevance feedback. In CBIR, one popular active learning for relevance feedback is the SVM active learning proposed by Tong et al [Tong and Chang 2001] [Tong and Chang 2001]. Some of its limitations have been addressed by some recent research work. For instance, to overcome the small sample learning issue, Wang et al. [Wang et al. 2003] proposed modifying the SVM active learning by engaging the unlabeled data with transductive SVM. Hoi et al. [Hoi and Lyu 2005] developed a more effective solution by combining semi-supervised learning techniques with supervised SVM active learning. Zhou et al. [Zhou et al. 2006] also proposed a co-training approach for combining semi-supervised learning and active learning for relevance feedback. Li et al. [Li et al. 2006] proposed a multitraining SVM method by adapting co-training techniques to CBIR. Despite the success, none of these studies address the batch mode active learning problem in which multiple unlabeled examples are selected in each iteration of active learning. A simple approach toward batch mode active learning is to select unlabeled examples close to decision boundary. However, as already point in [Dagli et al. 2006], the examples selected by this simple approach could be redundant, which leads to sub-optimal solutions. Several approaches have been proposed to address the batch sampling issue. Goh et al. [Goh et al. 2004; Panda et al. 2006] adopted the active learning method by incorporating the angular diversity measure, which was originally studied in machine learning community [Brinker 2003]. Dagli et al. [Dagli et al. 2006] recently proposed another similar approach using an information theoretic diversity measure approach and reported slightly better results than the angular diversity measure. However, our empirical results in this paper seem to somewhat different from their claims. This may be due to the difference of testbeds used. In this work, in contrast to previous heuristic

approaches for solving the batch sampling problem, we formally formulate this problem in a min-max learning framework, and propose two novel and effective algorithms to solve the optimization problems.

In addition to the research work in multimedia information retrieval, our work is also related to two broad research topics in machine learning: semi-supervised learning and active learning. In contrast to traditional supervised learning, semi-supervised learning exploits both labeled and unlabeled data, an approach which has been actively studied in recent years [Chapelle et al. 2006]. We investigate here the semi-supervised SVM technique [Sindhwani et al. 2005] with applications to relevance feedback in CBIR for solving the problem of learning with small number of labeled examples. On the other hand, active learning has been extensively studied in machine learning in the past decade [Cohn et al. 1995; Liere and Tadepalli 1997; McCallum and Nigam 1998; Schohn and Cohn 2000; Tong and Koller 2000]. However, traditional approaches often choose only one example for labeling in each active learning iteration and seldom explicitly address the batch sampling issue. Recently, some work has emerged on studying batch mode active learning [Hoi et al. 2006; Hoi et al. 2006; Yuhong Guo 2007]. But most of these solutions were developed under the probabilistic framework of kernel logistic regressions, which is not directly applicable to the SVM models. Our batch mode active learning technique in this work is motivated and built under the same theoretical framework used for SVMs.

3. SEMI-SUPERVISED SVM BATCH MODE ACTIVE LEARNING

In this section, we first formulate relevance feedback in CBIR as a problem of batch mode active learning, followed by the presentation of a semi-supervised kernel learning approach and the min-max framework for SVM batch mode active learning.

3.1 Preliminaries

Let us denote by $\mathcal{L} = \{(\mathbf{x}_1, y_i) \ldots, (\mathbf{x}_l, y_l)\}$ a set of l labeled image examples that are solicited through relevance feedback, and by $\mathcal{U} = \{\mathbf{x}_{l+1}, \ldots, \mathbf{x}_n\}$ a set of $n-l$ unlabeled image examples, where $\mathbf{x}_i \in \mathbb{R}^d$ represents an image by a d-dimensional vector.

We first formulate the relevance feedback of a CBIR system as an active learning problem. Let $\mathcal{S}$ be a set of k unlabeled image examples to be selected in relevance feedback, and $risk(f, \mathcal{S}, \mathcal{L}, \mathcal{U})$ be a risk function that depends on the classifier f, the labeled data $\mathcal{L}$, the unlabeled data $\mathcal{U}$, and the selected unlabeled examples $\mathcal{S}$ for relevance judgments. We chose $\mathcal{S}$ by minimizing the risk function $risk(f, \mathcal{S}, \mathcal{L}, \mathcal{U})$, which leads to the following combinatorial optimization problem:

$$S^* = \underset{S \subseteq \mathcal{U} \wedge |S|=k}{\arg\min} \; risk(f, \mathcal{S}, \mathcal{L}, \mathcal{U}) \tag{1}$$

We refer to the above problem as "*batch mode active learning*" because it selects multiple examples simultaneously. We emphasize that solving the problem in (1) is challenging since it is in general an NP-hard problem. This is in contrast to the conventional active learning where a single example is selected in each iteration of active learning.

We briefly review the basics of SVM since our study is focused on applying SVM for batch mode active learning. The key idea of SVM is to learn an optimal hyperplane that separates training examples with the maximal margin [Vapnik 1998]. A linear SVM finds

an optimal hyperplane $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$ by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{\lambda}{2}||\mathbf{w}||^2 + \sum_{i=1}^{l} \xi_i \\ s.t. \quad & y_i(\mathbf{w}_i^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \ldots, l \end{aligned} \tag{2}$$

where λ is the regularization parameter and ξ_is are slack variables that are introduced for the nonseparable examples. Kernel tricks are often used to extend the linear SVM in (2) to the nonlinear case, i.e.,

$$\min_{f \in \mathcal{H}_K} \sum_{i=1}^{l} \max(0, 1 - y_i f(\mathbf{x}_i)) + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}_K} \tag{3}$$

where $\mathcal{H}_K$ is the Hilbert space reproduced by a kernel function K. As indicated in (3), one of the key issue with the kernel SVM is to design an appropriate kernel function, which will be discussed in the following subsection.

3.2 A Semi-Supervised Support Vector Machine

Conventional SVM active learning relies on a supervised SVM model to train classifier $f(x)$ from labeled examples [Tong and Koller 2000; Tong and Chang 2001]. Supervised SVM models are often sensitive to the number of training examples and could deliver a poor performance when the number of labeled examples is small. We address this problem by exploiting a semi-supervised learning technique that learns a classifier from both labeled and unlabeled data.

Semi-supervised learning has been actively studied in recent years ([Chapelle et al. 2006] and references therein). In this work, we employ a unified kernel learning approach for semi-supervised learning [Hoi et al. 2006; Zhang and Ando 2005]. It first learns a data-dependent kernel from both labeled and unlabeled data, and then trains a supervised SVM model using the learned kernel function. Compared to the other SSL approaches, the unified kernel learning scheme is advantageous in its computational efficiency because the framework is divided into two independent stages, i.e., one stage for unsupervised kernel learning and the other stage for supervised kernel classifier training. A kernel deformation principle is adopted to learn a data-dependent kernel function [Sindhwani et al. 2005]. Below we briefly review the kernel deformation principle in [Sindhwani et al. 2005].

Let $\mathcal{H}$ denote the original Hilbert space reproduced by the kernel function $k(\cdot, \cdot)$, and $\tilde{\mathcal{H}}$ denote the deformed Hilbert space. We assume the following relationship between the two Hilbert spaces, i.e.,

$$\langle f, g \rangle_{\tilde{\mathcal{H}}} \; = \; \langle f, g \rangle_{\mathcal{H}} + \mathbf{f}^\top M \mathbf{g} \tag{4}$$

where $f(\cdot)$ and $g(\cdot)$ are two functions. $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))$ and $\mathbf{g} = (g(\mathbf{x}_1), \ldots, g(\mathbf{x}_n))$ evaluate functions $f(\cdot)$ and $g(\cdot)$ for both labeled and unlabeled examples, and M is the distance metric that captures the geometry relationship among all the data points. The deformation term in (4), i.e., $\mathbf{f}^\top M \mathbf{g}$, is introduced to assess the relationship between the function $f(\cdot)$ and $g(\cdot)$ based on the observed data points. Based on the above assumption in (4), [Sindhwani et al. 2005] derived the new kernel function $\tilde{k}(\cdot, \cdot)$ associated with the deformed space $\tilde{\mathcal{H}}$, i.e.,

$$\tilde{k}(\mathbf{x}, \mathbf{y}) = k(\mathbf{x}, \mathbf{y}) - \boldsymbol{\kappa}_{\mathbf{y}}^\top (I + MK)^{-1} M \boldsymbol{\kappa}_{\mathbf{x}} \tag{5}$$

where $K = [k(\mathbf{x}_i, \mathbf{x}_j)]_{n\times n}$ is the original kernel matrix for all the data points, and $\boldsymbol{\kappa}_{\mathbf{x}}$ is defined as $(k(\mathbf{x}_1, \mathbf{x}) \dots k(\mathbf{x}_n, \mathbf{x}))^\top$. To capture the geometrical structure of data, a common approach is to define M as a function of graph Laplacian L, i.e., $M = L$. Here, a graph Laplacian L is defined as $L = \text{diag}(S\mathbf{1}) - S$ where $S \in \mathbb{R}^{n\times n}$ is a similarity matrix and each element $S_{i,j}$ is calculated by an RBF function $\exp(-|\mathbf{x}_i - \mathbf{x}_j|_2^2/\sigma^2)$.
Remark To better understand the kernel deformation, we can rewrite (5) as follows:

$$\tilde{K} = K - K(I + MK)^{-1}MK = (K^{-1} + M)^{-1}$$

where $\tilde{K} = [\tilde{k}(\mathbf{x}_i, \mathbf{x}_j)]_{n\times n}$ is the kernel matrix computed by the new kernel function $\tilde{k}(\cdot,\cdot)$. As indicated by the above equation, the new kernel matrix $\tilde{K}$ can be viewed as the "reciprocal mean" of matrix K and M^{-1}. Hence, when we have a strong geometrical relationship among all the data points, namely M is "large", we expect the resulting new kernel matrix $\tilde{K}$ to be significantly deformed by the geometrical relationships in M.

Finally, for the remaining part of this article, notation K, instead of $\tilde{K}$, is used to refer to the kernel specified in (5), just for briefty.

3.3 SVM Batch Mode Active Learning

Conventional SVM active learning method employs the notion of *version space* for measuring the risk in active learning. Given training data $\mathcal{L}$ and a kernel function $k(\cdot,\cdot)$, the version space is defined as a set of hyperplanes that are able to separate training data from different classes in the feature space $\mathcal{H}_K$ induced by the kernel function $k(\cdot,\cdot)$. The optimal unlabeled example is found by maximizing the reduction in the volume of the version space. More details of SVM active learning can be found in [Tong and Koller 2000]. Although the above idea works well for selecting a single unlabeled example, it is difficult to extend it to select multiple examples because the number of partitions of version space increases exponentially in the number of selected examples. In the following subsections, we first present a new principle, termed "**min-max**" principle, for active learning, followed by the application of the min-max framework to batch mode active learning.

3.3.1 *Active Learning as Min-Max Optimization.* To motivate the min-max view of active learning, we first examine the SVM-based active learning for selecting single example, and show that it can be reformulated as a min-max optimization.

Let $g(f, \mathcal{L}, K)$ denote the margin-based objective function in the regularization framework in Eq. (3), i.e.,

$$g(f, \mathcal{L}, K) = \sum_{i=1}^{l} l(y_i, f(x_i)) + \frac{\lambda}{2}\|f\|_{\mathcal{H}_K}^2$$

where $l(y, \hat{y}) = \max(0, 1 - y\hat{y})$. The SVM-based active learning method [Tong and Koller 2000] selects the unlabeled example that is closest to the decision boundary. This can be expressed by the following optimization problem

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{U}} |f(\mathbf{x})| \tag{6}$$

The following theorem shows that the selection criterion in (6) is equivalent to a min-max formulation.

THEOREM 1. *The problem in (6) is equivalent to the following min-max optimization problem*

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{U}} \max_{y\in\{-1,+1\}} g(f, \mathcal{L}\cup(\mathbf{x}, y), K) \tag{7}$$

The proof of Theorem 1 can be found in Appendix A. The above analysis indicates that active learning can be viewed as a worst case analysis. In particular, to identify the most informative example, we select the unlabeled example $\mathbf{x}$ that minimizes the objective function $g(f, \mathcal{L}, K)$ regardless of its assigned class label y. The above analysis also allows us to identify the weakness of the SVM-based approach. In particular, when we measure the impact of an additional example $(\mathbf{x}, y)$ on the objective function $g(f, \mathcal{L}, K)$, we assume that the classifier f remains unchanged even with additional example $(\mathbf{x}, y)$. This is evidently an incorrect assumption, and will lead to an overestimation of the impact of $(\mathbf{x}, y)$ on the objective function. Hence, to address this problem, we remove the assumption of fixed classifier f, and propose to cast active learning as the following min-max optimization problem:

$$\arg\min_{\mathbf{x}\in\mathcal{U}} \max_{y\in\{-1,+1\}} \min_{f\in\mathcal{H}_K} g(f, \mathcal{L}\cup(\mathbf{x}, y), K) \tag{8}$$

It is important to note that by including the classifier f as part of min-max formulation, the unlabeled example selected by the above formulation will depart from the idea of selecting unlabeled examples that are close to decision boundary, which is key idea behind the SVM-based active learning. In the next subsection, we extend the formulation in (8) to SVM batch mode active learning that selects multiple examples in each round of learning.

3.3.2 *Min-max Framework for Batch Mode Active Learning.* To extend the min-max framework for batch mode active learning, we extend the problem in (8) to the following optimization problem:

$$\arg\min_{\mathcal{S}\subseteq\mathcal{U}\wedge|\mathcal{S}|=k} \max_{\mathbf{y}\in\{-1,+1\}^k} \min_{f\in\mathcal{H}_K} g(f, \mathcal{L}\cup(\mathcal{S}, \mathbf{y}), K) \tag{9}$$

where $\mathbf{y} = (y_1, \ldots, y_k)$ stands for the class labels assigned to the k selected examples in $\mathcal{S}$. Notation $(\mathcal{S}, \mathbf{y})$ is defined as

$$(\mathcal{S}, \mathbf{y}) = \{(\mathbf{x}_{i_j}, y_j), j = 1, \ldots, k | \mathbf{x}_{i_j} \in \mathcal{S}\}.$$

We emphasize that our objective, as specified in (9), is to find the unlabeled examples that will result in a smaller value for the SVM objective function $g(f, \mathcal{L}, K)$ regardless of the assigned class labels. Since the objective function of SVM is related to the generalization performance of test error, we believe the min-max criterion should essentially improve the generalization error effectively.

Before discussing the strategies for optimization, we devote the remaining part of this subsection to simplifying the optimization problem in (9).

First, we simplify the problem in (9) by removing the maximization with respect to $\mathbf{y}$. The result is summarized by the following theorem.

THEOREM 2. *The optimization problem in (9) is equivalent to the following problem:*

$$\arg\min_{\mathcal{S}\subseteq\mathcal{U}\wedge|\mathcal{S}|=k} \min_{f\in\mathcal{H}_K} \tilde{g}(f, \mathcal{L}, \mathcal{S}, K) \tag{10}$$

where

$$\tilde{g}(f, \mathcal{L}, \mathcal{S}, K) = \frac{\lambda}{2}\|f\|_{\mathcal{H}_K}^2 + \sum_{i=1}^{l} l(y_i, f(\mathbf{x}_i)) + \sum_{\mathbf{x}_j \in \mathcal{S}} |f(\mathbf{x}_j)| \tag{11}$$

The detailed proof can be found in Appendix B.

Next, we simplify the combinatorial optimization problem in (10) by replacing discrete variables with continuous ones. In particular, we introduce a continuous variable $q_i \in [0, 1]$ to represent the "degree" of selection for each unlabeled example in $\mathcal{U}$. This variable will replace the hard membership in (10). Since $q_i \in [0, 1]$, it can be viewed as some kind of probability of selecting an example for feedback. The following theorem shows a continuous version of the optimization problem in (10) using the probability q_i:

THEOREM 3. *The optimization problem in (10) is equivalent to the following optimization problem:*

$$\operatorname*{arg\,min}_{\mathbf{q}^\top \mathbf{1} = k, \mathbf{0} \preceq \mathbf{q} \preceq \mathbf{1}} \min_{f \in \mathcal{H}_K} \tilde{g}(f, \mathcal{L}, \mathbf{q}, K) \tag{12}$$

where

$$\tilde{g}(f, \mathcal{L}, \mathbf{q}, K) = \frac{\lambda}{2}\|f\|_{\mathcal{H}_K}^2 + \sum_{i=1}^{l} l(y_i, f(\mathbf{x}_i)) + \sum_{\mathbf{x}_j \in \mathcal{U}} q_j |f(\mathbf{x}_j)|$$

The detailed proof can be found in Appendix C.

Through the above derivation, we have arrived at (12), a substantially simpler problem compared to (9). In the next two subsections, we will discuss two approximate approaches that can solve the problem in (12) efficiently.

3.4 Approximate Approach (I): Quadratic Programming Approach for SVM Batch Mode Active Learning

Solving the optimization problem in (12) directly is challenging. The upper bound result in the following theorem allows us to simplify the optimization problem significantly.

THEOREM 4.

$$\min_{f \in \mathcal{H}_K} \tilde{g}(f, \mathcal{L}, \mathbf{q}, K) - \frac{k}{\lambda} \;\leq\; g(f^*, \mathcal{L}, K) + \frac{1}{\lambda}\mathbf{q}^\top \tilde{\mathbf{f}} + \frac{1}{2\lambda^2}\mathbf{q}^\top K_{u,u}\mathbf{q} \tag{13}$$

where $\tilde{\mathbf{f}} = (|f^*(\mathbf{x}_{l+1})|, \ldots, |f^*(\mathbf{x}_n)|)^\top$. *Function* $f^*(\mathbf{x})$ *is defined as*

$$f^* \;=\; \operatorname*{arg\,min}_{f \in \mathcal{H}_K} \; g(f, \mathcal{L}, K) \tag{14}$$

The details of the proof can be found in Appendix D.

Now, using the upper bound from Theorem 4 above, instead of optimizing the the objective function $\min_{f \in \mathcal{H}_K} \tilde{g}(f, \mathcal{L}, \mathbf{q}, K)$ directly, we can solve the problem by optimizing its upper bound, which leads to the following optimization problem for $\mathbf{q}$:

$$\begin{aligned} \min_{\mathbf{q} \in \mathbb{R}^{n-l}} \quad & \mathbf{q}^\top \tilde{\mathbf{f}} + \frac{1}{2\lambda}\mathbf{q}^\top K_{u,u}\mathbf{q} \\ \text{s. t.} \quad & \mathbf{q}^\top \mathbf{1} = k, \mathbf{0} \preceq \mathbf{q} \preceq \mathbf{1} \end{aligned} \tag{15}$$

where λ is a parameter introduced between the two terms. The above optimization is a standard quadratic programming (QP) problem that can be solved effectively by existing convex optimization software packages [Boyd and Vandenberghe 2004]. Finally, given the estimated q_i, we select the first k unlabeled examples with the largest probabilities q_i. Fig. 1 summarizes the overall algorithm ($\text{SVM}^{\text{SS(QP)}}_{\text{BMAL}}$) for semi-supervised SVM batch mode active learning with quadratic programming. It consists of two major steps: (a) learn a data-dependent kernel matrix $\tilde{K}$, and (b) train an SVM model with the kernel $\tilde{K}$ and find $\mathbf{q}$ by solving the optimization problem for batch mode active learning. Note that the first step can be done *offline* without knowing user queries, while the second step must be solved *online* for each individual query.

Remark I It is important to note that since (15) is only an APPROXIMATION of (12), therefore the optimal solution to (15) is no longer binary. We will come back to this issue when we present the second approximate strategy.

Remark II It is interesting to examine the meanings of the two terms in the objective function in (15). The first term, i.e., $\mathbf{q}^\top \tilde{\mathbf{f}}$, is related to the classification uncertainty. By minimizing $\mathbf{q}^\top \tilde{\mathbf{f}}$, we preferentially select examples close to the decision boundary. Meanwhile, the second term, $\mathbf{q}^\top K_{u,u}\mathbf{q}$, is related to the redundancy among the selected examples. By minimizing $\mathbf{q}^\top K_{u,u}\mathbf{q}$, the selected examples tend to have small similarity among themselves. This is consistent with our intuition that we should select the most uncertain and diversified examples for labeling by a batch mode active learning algorithm.

3.5 Approximate Approach (II): Combinatorial Optimization Algorithm for SVM Batch Mode Active Learning

Although Eq.(15) provides decent performance for batch mode active learning, it requires solving a quadratic programming problem, which could be computationally expensive when the number of unlabeled examples is large. In this subsection, we aim to directly address the binary selection problem with a simple yet rather effective greedy combinatorial optimization algorithm based on the theory of submodular functions.

Let $\mathcal{S}$ denote the collection of unlabeled examples that were selected for active learning. Then, the discrete version of Eq. (15) is written as

$$\min_{\mathcal{S}\subset\mathcal{U},|\mathcal{S}|=k} \sum_{i\in\mathcal{S}} \tilde{f}_i + \frac{\lambda}{2} \sum_{i,j\in\mathcal{S}} [K_{u,u}]_{i,j} \tag{16}$$

It is important to note the difference between the discrete version in (16) and the continuous version in (15). In particular, in the discrete version in (16), only the sub-matrix of K that involves the selected elements in $\mathcal{S}$ will contribute to the overall objective function. In contrast, the objective function in (15) involves all the elements in the kernel matrix K because of the soft memberships in $\mathbf{q}$. In this sense, the objective function in (16) is more accurate in identifying the selected examples than (16).

We further note that Eq.(16) is a combinatorial optimization problem, and is usually NP-hard. In order to efficiently solve the above problem, we will exploit the properties of submodular functions. Before we present our algorithm for Eq. (16), we will first give an overview the concept of submodular functions and its properties related to combinatorial optimization.

To define submodular functions, we consider functions of sets, denoted by $f(\mathcal{S})$ where $\mathcal{S}$ is a set. A set function $f(\mathcal{S})$ is called a submodular function if and only if the following

Algorithm 1 Semi-Supervised SVM Batch Mode Active Learning with QP ($\text{SVM}_{\text{BMAL}}^{\text{SS(QP)}}$)

INPUT:
$\mathcal{L}, \mathcal{U}$ /* labeled and unlabeled data */
l, n, k /* label size, total data size, batch size */
$\mathbf{K}$ /* an input kernel, e.g. an RBF kernel */
PARAMETERS:
λ /* batch mode active learning regularization parameter */
$\tilde{\mathbf{K}}$ /* a data-dependent kernel */
VARIABLES:
$\mathbf{q}$ /* probabilities of selecting unlabeled examples for labeling*/
OUTPUT:
$\mathcal{S}$ /* a batch of unlabeled examples selected for labeling*/
PROCEDURE
/* *Unsupervised kernel design procedure* (Offline)*/
1: Build a graph Laplacian from data $\mathbf{L} = \text{Laplacian}(\mathcal{L} \cup \mathcal{U})$;
2: Learn a data-dependent kernel $\tilde{\mathbf{K}}$ by Eq. (5);
/* *Start batch mode active learning procedure* (Online) */
1: Train an SVM classifier: $f^* = \mathsf{SVM\_Train}(\mathcal{L}, \tilde{\mathbf{K}})$; /* call a standard SVM solver */
2: Compute $\tilde{\mathbf{f}} = (|f^*(\mathbf{x}_{l+1})|, \ldots, |f^*(\mathbf{x}_n)|)^\top$;
3: $\mathbf{H} = \lambda\tilde{\mathbf{K}}$; $\mathbf{f} = \tilde{\mathbf{f}}$;
4: Aeq = $\mathbf{1}_{1\times u}$; beq = k;
5: $\mathbf{q} = \mathsf{quadprog}(\mathbf{H}, \mathbf{f}, \text{Aeq}, \text{beq}, \mathbf{0} \preceq \mathbf{q} \preceq \mathbf{1})$; /* call a standard QP solver */
6: $\mathcal{S} = \varnothing$;
7: **while** ($|\mathcal{S}| < k$) **do**
8: $\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{U}} q(\mathbf{x})$;
9: $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathbf{x}^*\}$; $\mathcal{U} \leftarrow \mathcal{U}\backslash\{\mathbf{x}^*\}$;
10: **end while**
11: **return** $\mathcal{S}$.
END

Fig. 1. Quadratic Programming (QP) approach for the proposed Semi-Supervised SVM Batch Mode Active Learning ($\text{SVM}_{\text{BMAL}}^{\text{SS(QP)}}$)

condition holds for any set $\mathcal{A} \subset \mathcal{B}$ and any element $e \notin \mathcal{B}$:

$$f(\mathcal{A} \cup e) - f(\mathcal{A}) \geq f(\mathcal{B} \cup e) - f(\mathcal{B}) \tag{17}$$

where we abbreviate $\mathcal{A} \cup \{e\}$ by $\mathcal{A} \cup e$. Given a submodular function $f(\mathcal{S})$ and the related combinatorial optimization problem, i.e.,

$$\max_{|S|=k} \; f(S), \tag{18}$$

a straightforward approach is to solve it by the following greedy approach: we start with an empty set for $\mathcal{S}$; in each iteration, we expand the set $\mathcal{S}$ with the element e that maximizes the difference $f(\mathcal{S} \cup e) - f(\mathcal{S})$. We keep on expanding $\mathcal{S}$ till the number of elements in $\mathcal{S}$ is k. The following theorem provides a performance guarantee for this greedy algorithm.

THEOREM 5. *[Nemhauser et al. 1978]. Consider the combinatorial optimization problem in (18). Let S^* denote the global optimal solution that solves (18), and $\hat{S}$ de-*

note the approximate solution found by the greedy algorithm. We have

$$f(\mathcal{S}) \geq f(\mathcal{S}^*)(1 - 1/C_e)$$

if $f(\mathcal{S})$ satisfies the following conditions:

(1) $f(\mathcal{S})$ is a nondecreasing function, namely $f(\mathcal{A}) \leq f(\mathcal{B})$ if $\mathcal{A} \subset \mathcal{B}$,
(2) $f(\mathcal{S})$ is a submodular function, and
(3) $f(\emptyset) = 0$

Here, C_e refers to the natural exponential.

In order to fully explore Theorem 5, we need to convert the problem in (16) into a maximization problem with the objective function that satisfies the three criteria stated in Theorem 5. To this end, we define the following objective function for maximization:

$$\begin{aligned} g(\mathcal{S}) &= \sum_{i \in \mathcal{S}} (\tilde{f}_0 - \tilde{f}_i) + \frac{\lambda}{2} \left(\sum_{i \in S} \theta\sqrt{n} - \sum_{i,j \in S} (1 - \delta_{i,j}) [K_{u,u}]_{i,j} \right) \\ &= |\mathcal{S}| \left(\tilde{f}_0 + \frac{\lambda}{2} \theta\sqrt{n} \right) + \frac{\lambda}{2} \sum_{i \in \mathcal{S}} [K_{u,u}]_{i,i} - \left(\mathbf{q}_{\mathcal{S}}^\top \tilde{\mathbf{f}} + \frac{\lambda}{2} \mathbf{q}_{\mathcal{S}}^\top K_{u,u} \mathbf{q}_{\mathcal{S}} \right) \end{aligned} \quad (19)$$

where

$$\tilde{f}_0 = \max_{1 \leq i \leq n} \tilde{f}_i, \quad \theta = \text{tr}(K_{u,u}), \quad (20)$$

and δ and $\mathbf{q}_{\mathcal{S}}$ are respectively defined as follows:

$$\delta_{i,j} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases} \qquad [\mathbf{q}_{\mathcal{S}}]_i = \begin{cases} 1 & \text{if } i \in \mathcal{S}, \\ 0 & \text{otherwise.} \end{cases} \quad (21)$$

Let's compare (19) with the objective function in (15). When compared to (15), two additional terms are introduced in (19), i.e., $|\mathcal{S}| \left(\tilde{f}_0 + \frac{\lambda}{2} \theta\sqrt{n} \right)$ and $\frac{\lambda}{2} \sum_{i \in \mathcal{S}} [K_{u,u}]_{i,i}$. It will later be revealed in Theorem 6 that it is these two terms that ensures (19) is a submodular function, which makes it possible to apply the result in Theorem 5. Furthermore, when the number of selected examples is fixed (i.e., $|\mathcal{S}|$ is a constant) and the self kernel similarity is constant (i.e., $[K_{u,u}]_{i,i}$ is constant for any example x_i)[1], the first two terms in (19) are independent from the selected examples $\mathcal{S}$. As a result, maximizing $g(\mathcal{S})$ in (19) is equivalent to the minimization problem in (15). Hence, in the following discussion, we focus on the problem of maximizing $g(\mathcal{S})$, i.e.,

$$\max_{|\mathcal{S}|=k} g(\mathcal{S}) = k \left(\tilde{f}_0 + \frac{\lambda}{2} \theta\sqrt{n} \right) + \frac{\lambda}{2} \sum_{i \in \mathcal{S}} [K_{u,u}]_{i,i} - \left(\mathbf{q}_{\mathcal{S}}^\top \tilde{\mathbf{f}} + \frac{\lambda}{2} \mathbf{q}_{\mathcal{S}}^\top K_{u,u} \mathbf{q}_{\mathcal{S}} \right) \quad (22)$$

A simple approach for the above optimization problem is the greedy approach. At the tth iteration, we denote by $\mathcal{S}_t$ the set of selected examples for the current iteration. The next example is chosen to maximize $g(\mathcal{S})$, which is equivalent to the following problem:

$$j^* = \min_{j \notin \mathcal{S}_t} h(j; \mathcal{S}_t) \quad (23)$$

[1] An example of such a kernel is RBF kernel

where

$$h(j; \mathcal{S}_t) = \tilde{f}_j + \lambda \sum_{i \in \mathcal{S}_t} [K_{u,u}]_{i,j} \tag{24}$$

Fig. 2 summarizes the proposed greedy Combinatorial Optimization (CO) algorithm for semi-supervised SVM batch mode active learning ($\text{SVM}_{\text{BMAL}}^{\text{SS(CO)}}$). The following theorem provides the performance guarantee for the proposed algorithm in Fig. 2.

THEOREM 6. *Assume all elements in the kernel matrix are non-negative, i.e.,* $[K_{u,u}]_{i,j} \geq 0$ *for any* i *and* j*. Let* $\hat{\mathcal{S}}$ *denote the set found by the greedy CO algorithm in Fig. 2, and* $\mathcal{S}^*$ *denote the optimal set that solves the problem in (22). We have the following performance guarantee:*

$$\frac{g(\hat{\mathcal{S}})}{g(\mathcal{S}^*)} \geq 1 - \frac{1}{C_e}$$

The key to proving the above theorem is to show that $g(\mathcal{S})$ defined in (19) satisfies the three conditions specified in Theorem 5. The details of the proof can be found in Appendix E.

4. EXPERIMENTAL RESULTS

4.1 Overview

To evaluate the performance of the proposed algorithm, we conduct an extensive set of CBIR experiments by comparing the proposed algorithm to several state-of-the-art active learning methods that have been used in image retrieval. Specifically, we design the experiments to evaluate two major factors that could significantly affect the results of batch mode active learning within the context of CBIR:

(1) **label size**, i.e., the number of labeled images judged by a user in the first around of image retrieval when no relevance feedback is applied;

(2) **batch size**, i.e., the number of data examples to be selected for labeling by active learning in each iteration of relevance feedback.

4.2 Experimental Testbed and Feature Extraction

Two benchmark CBIR datasets are used in our experiments [2]:(1) COREL photo images [Hoi et al. 2006], and (2) ImageCLEF medical images [Muller et al. 2007].

4.2.1 *COREL Photo Image Dataset.* For COREL images, we form a dataset that contains $5,000$ images from 50 different categories. Each category consists of exactly 100 images that are randomly selected from relevant examples in the COREL database. Every category represents a different semantic topic, such as *antelope*, *butterfly*, *car*, *cat*, *dog*, *horse* and *lizard*. Figure 3 (a) shows some image examples in this dataset.

For feature representation on this testbed, we extract three types of features. (1) *Color*: For each 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 in our testbed. (2) *Edge*: An edge direction histogram is extracted for each image. Each image is converted into a gray image, and a Canny edge detector is applied to

[2]The datasets are available at `http://www.cais.ntu.edu.sg/~chhoi/SVMBMAL/`

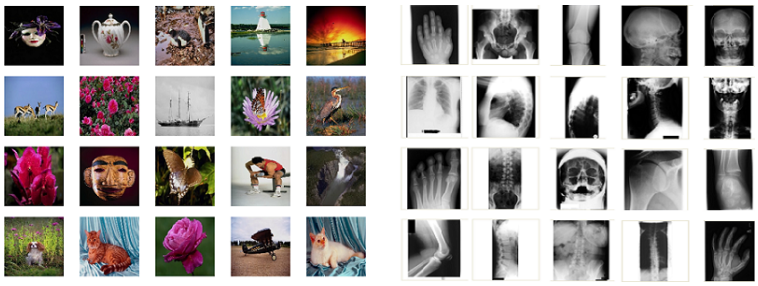
Algorithm 2 Semi-Supervised SVM Batch Mode Active Learning with CO ($\mathrm{SVM}_{\mathrm{BMAL}}^{\mathrm{SS(CO)}}$)
INPUT:
$\mathcal{L}, \mathcal{U}$ /* labeled and unlabeled data */
l, n, k /* label size, total data size, batch size */
$\mathbf{K}$ /* an input kernel, e.g. an RBF kernel */
PARAMETERS:
λ /* batch mode active learning regularization costs */
$\tilde{\mathbf{K}}$ /* a data-dependent kernel */
VARIABLES:
$\mathbf{h}$ /* cost function of selecting unlabeled examples for labeling*/
OUTPUT:
$\mathcal{S}$ /* a batch of unlabeled examples selected for labeling*/
PROCEDURE
/* *Unsupervised kernel design procedure* (Offline)*/
1: Build a graph Laplacian from data $\mathbf{L} = \mathrm{Laplacian}(\mathcal{L} \cup \mathcal{U})$;
2: Learn a data-dependent kernel $\tilde{\mathbf{K}}$ by Eq. (5);
/* *Start batch mode active learning procedure* (Online) */
1: Train an SVM classifier: $f^* = \mathsf{SVM\_Train}(\mathcal{L}, \tilde{\mathbf{K}})$; /* call a standard SVM solver */
2: Compute $\hat{\mathbf{f}} = (|f^*(\mathbf{x}_{l+1})|, \ldots, |f^*(\mathbf{x}_n)|)^\top$;
3: $\mathcal{S} = \varnothing$;
4: **while** ($|\mathcal{S}| < k$) **do**
5: **for** each $\mathbf{x}_j \in \mathcal{U}$ **do**
6: $h(\mathbf{x}_j) = \hat{f}(\mathbf{x}_j) + \lambda \sum_{\mathbf{x}_i \in \mathcal{S}_t} [\tilde{K}_{u,u}]_{i,j}$;
7: **end for**
8: $\mathbf{x}_j^* = \arg\max_{\mathbf{x}_j \in \mathcal{U}} h(\mathbf{x}_j)$;
9: $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathbf{x}_j^*\}$; $\mathcal{U} \leftarrow \mathcal{U} \backslash \{\mathbf{x}_j^*\}$;
10: **end while**
11: **return** $\mathcal{S}$.
END

Fig. 2. The greedy Combinatorial Optimization (CO) approach for the proposed Semi-Supervised SVM Batch Mode Active Learning ($\mathrm{SVM}_{\mathrm{BMAL}}^{\mathrm{SS(CO)}}$)

obtain the edges, from which the edge direction histogram is computed. The edge direction histogram is quantized into 18 bins of 20 degrees each, thus a total of 18 edge features are extracted. (3) *Texture*: The Discrete Wavelet Transformation (DWT) is performed on the gray images. Each wavelet decomposition on a gray 2D-image results in four scaled-down subimages. In total, 3-level decomposition is conducted and features are extracted from 9 of the subimages by computing entropy. Thus, a 9-dimensional wavelet vector is used. Thus, in total, a 36-dimensional feature vector is used to represent each image.

4.2.2 *ImageCLEF Medical Image Dataset.* For ImageCLEF medical images, we form a 20-category Dataset that contains $6,157$ images from 20 semantic categories. Each category consists at least 100 medical images from ImageCLEF [Muller et al. 2007], which are either x-ray or CT images. Every category represents a different semantic topic, such as *chest*, *cranium*, *hand*, *cervical spine*, *foot*, and *pelvis*. Figure 3 (b) shows some image examples in this dataset.

For feature representation on this dataset, we only consider the texture features, as most medical images are gray images. To this purpose, we extract the Gabor feature [Manju-



(a) COREL Photo image dataset (b) ImageCLEF medical image dataset

Fig. 3. Some image samples from the two image datasets used in our experiments.

nath and Ma 1996], which captures the local structures corresponding to different spatial frequencies (scales), spatial localizations, and orientations. For each image, we apply the Gabor wavelet transformation with 5 scale levels and 8 orientations, which results in a total of 40 subimages for the input image. We then calculate three statistical moments to represent the texture features, including mean, variance, and skewness. In total, a 120-dimensional Gabor vector is used to represent a medical image.

4.3 Compared Schemes and Experimental Setup

In the experiments, we compare a number of state-of-the-art algorithms for active learning in CBIR. The compared algorithms include the following existing algorithms:

(1) Random: the simplest and naive approach for relevance feedback with SVM [Tong and Koller 2000], denoted by Random.

(2) SVM Active Learning: the baseline is the original SVM active learning algorithm that samples examples closest to the decision boundary [Tong and Chang 2001], denoted by $\mathrm{SVM}_{\mathrm{AL}}$.

(3) SVM Active Learning with Angular Diversity: a heuristic modification of SVM active learning that incorporates diversity in batch sampling [Brinker 2003], in which the diversity measure is based on the cosine value of the maximum angle with a set of induced hyperplanes. We denote it by $\mathrm{SVM}_{\mathrm{AL}}^{\mathrm{DIVA}}$.

(4) SVM Active Learning with Entropic Diversity: similar to (2), a recently proposed active learning method that incorporates diversity for active learning [Dagli et al. 2006], which employed an information-theoretic approach for diversity measure. We denote it by $\mathrm{SVM}_{\mathrm{AL}}^{\mathrm{DIVE}}$.

(5) Semi-Supervised Active Learning: a fusion of semi-supervised learning and SVM active learning, intended to overcome the small sample learning issue of regular SVM active learning [Hoi and Lyu 2005], denoted by SSAL.

and four variants of our proposed batch mode active learning (BMAL) algorithms:

(6) SVM BMAL with Quadratic Programming: the proposed BMAL method solved by the quadratic programming algorithm with the supervised SVM method, denoted by $\mathrm{SVM}_{\mathrm{BMAL}}^{(\mathrm{QP})}$.

(7) SVM BMAL with Combinatorial Optimization: the proposed BMAL method solved by the combinatorial optimization algorithm with the supervised SVM method, denoted by $\mathrm{SVM}_{\mathrm{BMAL}}^{(\mathrm{CO})}$.

(8) Semi-Supervised SVM BMAL with Quadratic Programming: the proposed semi-supervised SVM BMAL method solved by the quadratic programming algorithm, denoted by $\mathrm{SVM}_{\mathrm{BMAL}}^{\mathrm{SS(QP)}}$.

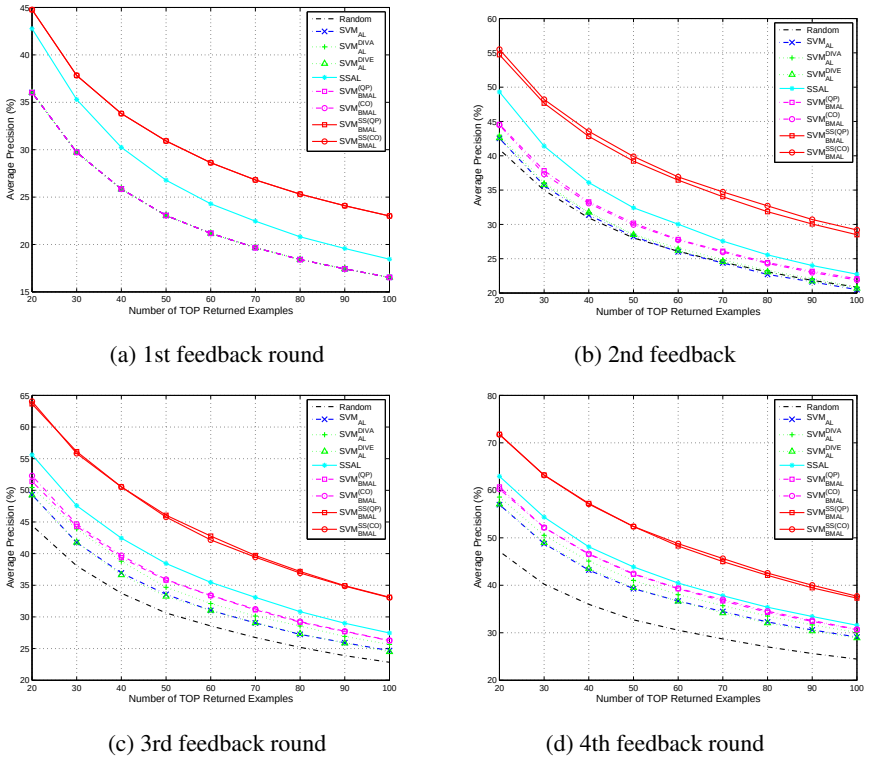
(9) Semi-Supervised SVM BMAL with Combinatorial Optimization: the proposed semi-supervised SVM BMAL method solved by the combinatorial optimization algorithm, denoted by $\mathrm{SVM}_{\mathrm{BMAL}}^{\mathrm{SS(CO)}}$.

To evaluate the average performance, we conduct every experiment by a set of 200 random queries with image examples sampled from the datasets. We simulate the CBIR procedure by returning the l images with shortest Euclidean distance to a given query example. The retrieved l images are then labeled and used as the set of initially labeled data to train the relevance feedback algorithms. An RBF kernel with fixed kernel width is used for all the algorithms. Regarding the parameter setting, the regularization parameter λ is set to 0.01 (or $C = 100$) for SVM in all experiments, and the λ set to 1 for both proposed semi-supervised batch mode active learning algorithms (i.e., $\mathrm{SVM}_{\mathrm{BMAL}}^{\mathrm{SS(QP)}}$ and $\mathrm{SVM}_{\mathrm{BMAL}}^{\mathrm{SS(CO)}}$). The combination parameters used in the two diversity-based active learning methods $\mathrm{SVM}_{\mathrm{AL}}^{\mathrm{DIVA}}$ and $\mathrm{SVM}_{\mathrm{AL}}^{\mathrm{DIVE}}$ are tuned by cross validation using a holdout set. For performance evaluation metrics, average precision (AP) and average recall (AR) are adopted, in which the relevance judgements are based on whether the query image and the retrieved image belong to the same category. The same evaluation methodology has been widely adopted in previous CBIR research [Tong and Chang 2001; Hoi and Lyu 2005]. Finally, we implement the proposed algorithms and other compared methods all in MATLAB and evaluated their performances on a Windows PC with Dual-Core 3.4GHz CPU and 3GB RAM. Because of limited space, in the following subsections, we focus on the methods' quantitative performance. More results on visual retrieval comparison are available online `http://www.cais.ntu.edu.sg/~chhoi/SVMBMAL/`.

4.4 Experiment I: Fixed Label Size and Batch Size

We first conduct experiments with both *label size* and *batch size* fixed to 10. Figure 4 and Figure 5 show the average precision for the first four rounds of relevance feedback on both datasets, respectively. In these figures, the black line represents the random method, the blue line represents the baseline $\mathrm{SVM}_{\mathrm{AL}}$ method, the two green dotted lines are $\mathrm{SVM}_{\mathrm{AL}}^{\mathrm{DIVA}}$ and $\mathrm{SVM}_{\mathrm{AL}}^{\mathrm{DIVE}}$, the cyan solid line is SSAL, the two pink dotted lines are the two proposed BMAL algorithms with supervised SVMs $\mathrm{SVM}_{\mathrm{BMAL}}^{(\mathrm{QP})}$ and $\mathrm{SVM}_{\mathrm{BMAL}}^{(\mathrm{CO})}$ and the two red solid lines are the two proposed BMAL algorithms with semi-supervised SVMs $\mathrm{SVM}_{\mathrm{BMAL}}^{\mathrm{SS(QP)}}$ and $\mathrm{SVM}_{\mathrm{BMAL}}^{\mathrm{SS(CO)}}$, respectively.

Several observations can be drawn from the results. First, we observe that all the eight active learning methods outperform the baseline random method across all the iterations for both datastes. This result indicates that all the active learning methods are indeed working well. Second, we observe that through all the iterations, for both datasets, the four active learning methods that exploit semi-supervised learning techniques (i.e., SSAL, $\mathrm{SVM}_{\mathrm{BMAL}}^{\mathrm{SS(CO)}}$, and $\mathrm{SVM}_{\mathrm{BMAL}}^{\mathrm{SS(QP)}}$) outperform the other six methods in comparison that do not utilize unlabeled data. This is further illustrated by comparing the proposed algorithms to their counterparts that do not utilize the unlabeled data. We also observe that without the assistance of semi-supervised learning, the two proposed algorithms for batch mode active learning performs considerably worse than SSAL; however, with the help of semi-supervise learning, we notice a very significant improvement in the batch mode active learning. All these results indicate the importance of combining semi-supervised learning techniques with active learning methods. Third, we observe that the two proposed algorithms, i.e., $\mathrm{SVM}_{\mathrm{BMAL}}^{\mathrm{SS(CO)}}$ and $\mathrm{SVM}_{\mathrm{BMAL}}^{\mathrm{SS(QP)}}$, outperform all the algorithms in comparison.



(a) 1st feedback round

(b) 2nd feedback

(c) 3rd feedback round

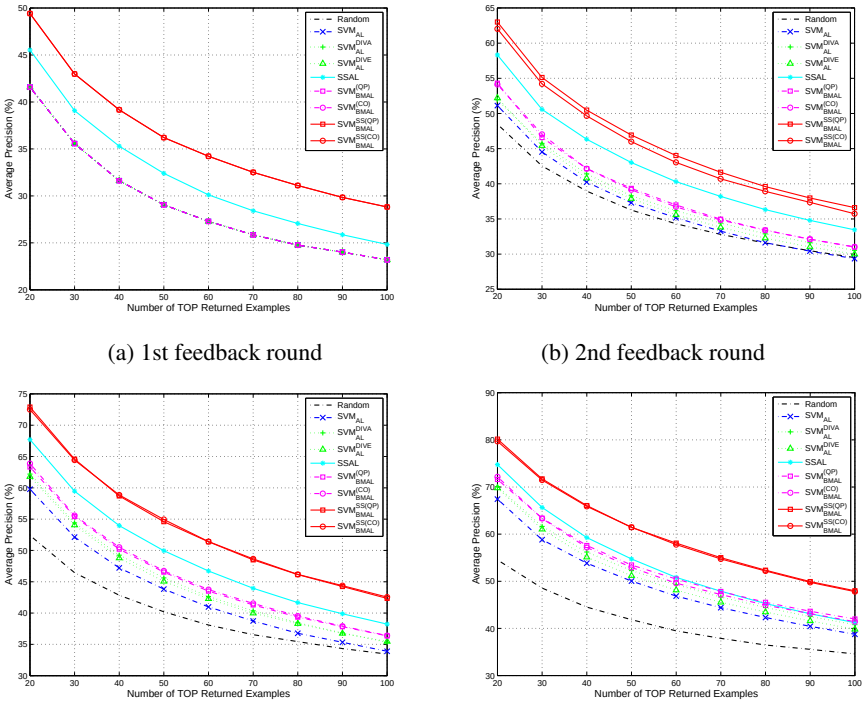
(d) 4th feedback round

Fig. 4. Performance of several active learning algorithms with fixed label and batch sizes on the COREL image testbed.

In particular, the two proposed algorithms outperform SSAL, the third best algorithm, with a considerable margin. Since the two proposed algorithms distinguish from SSAL in that they are designed for batch mode active learning while SSAL does not, we thus conclude the importance of batch mode active learning when multiple examples are selected in each iteration. Finally, comparing the two proposed batch mode active learning methods, we found they have similar performance. For most cases, they perform almost the same except for the second iteration, where $\text{SVM}^{\text{SS(CO)}}_{\text{BMAL}}$ achieves slightly better performance on the COREL dataset while $\text{SVM}^{\text{SS(QP)}}_{\text{BMAL}}$ performs better on the ImageCLEF dataset.

4.5 Experiment II: Varied Label Size

The second set of experiments is to evaluate the performance with varied label sizes. Table I and Table II show the results of average precision for the top 20 returned images with one active learning iteration for both datasets obtained by varying the label size and fixing the batch size to 10. In the tables, "MAP" and "MAR" stand for Mean Average Precision and Mean Average Recall, respectively. Note that, due to the space limitation, we omit the results for $\text{SVM}^{\text{(CO)}}_{\text{BAML}}$ and $\text{SVM}^{\text{(QP)}}_{\text{BMA}}$, the two variants of the proposed algorithms that do not exploit unlabeled data. This is because their performance is significantly worse than



(a) 1st feedback round

(b) 2nd feedback round

(c) 3rd feedback round

(d) 4th feedback round

Fig. 5. Performance of several active learning algorithms with fixed label and batch sizes on the ImageCLEF testbed.

Label Size	$\mathrm{SVM_{AL}}$	$\mathrm{SVM_{AL}^{DIVA}}$	$\mathrm{SVM_{AL}^{DIVE}}$	SSAL	$\mathrm{SVM_{BMAL}^{SS(QP)}}$	$\mathrm{SVM_{BMAL}^{SS(CO)}}$
5	0.365	0.372 + 1.8 %	0.366 + 0.2 %	0.426 + 16.8 %	**0.484** + 32.7 %	0.481 + 31.8 %
10	0.425	0.430 + 1.1 %	0.426 + 0.2 %	0.493 + 15.9 %	0.547 + 28.7 %	**0.555** + 30.5 %
15	0.478	0.492 + 2.9 %	0.489 + 2.2 %	0.557 + 16.5 %	**0.607** + 26.9 %	0.604 + 26.4 %
20	0.548	0.550 + 0.3 %	0.549 + 0.1 %	0.600 + 9.6 %	**0.651** + 18.9 %	0.642 + 17.2 %
25	0.592	0.599 + 1.1 %	0.590 - 0.3 %	0.642 + 8.4 %	0.681 + 15.0 %	**0.682** + 15.3 %
30	0.616	0.627 + 1.9 %	0.612 - 0.7 %	0.667 + 8.3 %	**0.700** + 13.7 %	0.696 + 13.0 %
MAP	0.504	0.511 + 1.5 %	0.505 + 0.2 %	0.564 + 11.9 %	**0.612** + 21.4 %	0.610 + 21.0 %

Table I. The *Average Precision* performance of the top 20 returned results with different **Label Sizes** on the COREL image testbed.

Label Size	$\mathrm{SVM_{AL}}$	$\mathrm{SVM_{AL}^{DIVA}}$	$\mathrm{SVM_{AL}^{DIVE}}$	SSAL	$\mathrm{SVM_{BMAL}^{SS(QP)}}$	$\mathrm{SVM_{BMAL}^{SS(CO)}}$
5	0.440	0.454 + 3.2 %	0.445 + 1.1 %	0.509 + 15.6 %	0.544 + 23.6 %	**0.554** + 25.9 %
10	0.511	0.520 + 1.6 %	0.522 + 2.1 %	0.583 + 14.1 %	**0.630** + 23.2 %	0.620 + 21.3 %
15	0.579	0.568 - 1.8 %	0.570 - 1.5 %	0.629 + 8.6 %	0.659 + 13.9 %	**0.664** + 14.7 %
20	0.608	0.626 + 3.0 %	0.628 + 3.3 %	0.644 + 6.0 %	0.677 + 11.4 %	**0.687** + 12.9 %
25	0.654	0.665 + 1.6 %	0.666 + 1.7 %	0.678 + 3.6 %	**0.712** + 8.8 %	0.709 + 8.3 %
30	0.666	0.681 + 2.3 %	0.684 + 2.7 %	0.702 + 5.4 %	0.730 + 9.6 %	**0.737** + 10.6 %
MAP	0.576	0.586 + 1.6 %	0.586 + 1.6 %	0.624 + 8.3 %	0.659 + 14.3 %	**0.662** + 14.8 %

Table II. The *Average Precision* performance of the top 20 returned results with different **Label Sizes** on the ImageCLEF medical image testbed.

the two proposed algorithms, as already demonstrated in the previous subsection.

Label Size	$\mathrm{SVM_{AL}}$	$\mathrm{SVM_{AL}^{DIVA}}$	$\mathrm{SVM_{AL}^{DIVE}}$	SSAL	$\mathrm{SVM_{BMAL}^{SS(QP)}}$	$\mathrm{SVM_{BMAL}^{SS(CO)}}$
5	0.191	0.193 + 1.0 %	0.189 - 1.3 %	0.208 + 8.6 %	**0.251** + 31.2 %	0.248 + 29.8 %
10	0.205	0.209 + 1.9 %	0.207 + 0.9 %	0.227 + 11.0 %	0.285 + 39.0 %	**0.292** + 42.5 %
15	0.219	0.225 + 2.6 %	0.222 + 1.1 %	0.252 + 14.7 %	0.301 + 37.2 %	**0.302** + 37.4 %
20	0.253	0.261 + 3.1 %	0.260 + 2.5 %	0.288 + 13.6 %	**0.333** + 31.4 %	0.332 + 31.2 %
25	0.279	0.285 + 2.0 %	0.277 - 0.7 %	0.306 + 9.5 %	0.348 + 24.5 %	**0.353** + 26.3 %
30	0.294	0.302 + 2.7 %	0.293 - 0.5 %	0.325 + 10.5 %	0.364 + 23.8 %	**0.365** + 24.0 %
MAP	0.240	0.246 + 2.3 %	0.241 + 0.3 %	0.267 + 11.3 %	0.314 + 30.5 %	**0.315** + 31.1 %

Table III. The *Average Recall* performance of the top 100 returned results with different **Label Sizes** on the COREL image testbed.

From the results in both Tables, we observe first that the two diversity-based active learning methods $\mathrm{SVM_{AL}^{DIVA}}$ and $\mathrm{SVM_{AL}^{DIVE}}$ achieve no more than 4% improvement over the baseline. In contrast, SSAL achieves considerably better performance with 4% to 16% improvement over the baseline. The two proposed algorithms achieve the best results on both datasets, with improvements almost double that of SSAL. Comparing the two proposed algorithms, we found that their performances are very close; the difference in their overall improvements over the baseline is smaller than 0.5%.

In addition, we found that the average improvement is reduced when the size of initially labeled images becomes larger. For example, on the COREL dataset, the relative improvement made by the proposed $\mathrm{SVM_{BMAL}^{SS(CO)}}$ algorithm is 30.5% when the label size is 10, and is reduced to 13.0% when the label size is 30. This again shows that the proposed method is able to effectively address the problem of small training size. Finally, we also show

Label Size	$\mathrm{SVM_{AL}}$	$\mathrm{SVM_{AL}^{DIVA}}$	$\mathrm{SVM_{AL}^{DIVE}}$	SSAL	$\mathrm{SVM_{BMAL}^{SS(QP)}}$	$\mathrm{SVM_{BMAL}^{SS(CO)}}$
5	0.086	0.088 + 3.0 %	0.086 + 0.5 %	0.097 + 13.6 %	0.107 + 24.5 %	**0.108** + 25.7 %
10	0.097	0.103 + 5.8 %	0.102 + 4.7 %	0.108 + 10.9 %	**0.122** + 25.9 %	0.118 + 21.6 %
15	0.109	0.110 0.4 %	0.109 + 0.0 %	0.120 + 9.7 %	0.126 + 15.5 %	**0.128** + 16.7 %
20	0.114	0.115 + 0.7 %	0.116 + 1.7 %	0.121 + 5.9 %	0.131 + 14.5 %	**0.134** + 16.9 %
25	0.122	0.123 + 1.1 %	0.122 + 0.3 %	0.128 + 5.0 %	**0.138** + 13.7 %	0.137 + 13.1 %
30	0.123	0.126 + 2.2 %	0.127 + 3.2 %	0.133 + 7.9 %	**0.142** + 14.8 %	**0.142** + 15.0 %
MAR	0.109	0.111 + 2.1 %	0.110 + 1.7 %	0.118 + 8.5 %	0.128 + 17.6 %	**0.128** + 17.7 %

Table IV. The *Average Recall* performance of the top 100 returned results with different **Label Sizes** on the ImageCLEF medical image testbed.

the average recall results for the top 100 returned images on both datasets respectively in Table III and Table IV; the observations are similar to those for average precision, further validating the advantages of the proposed algorithms as compared to the others.

Batch Size	$\mathrm{SVM_{AL}}$	$\mathrm{SVM_{AL}^{DIVA}}$	$\mathrm{SVM_{AL}^{DIVE}}$	SSAL	$\mathrm{SVM_{BMAL}^{SS(QP)}}$	$\mathrm{SVM_{BMAL}^{SS(CO)}}$
5	0.492	0.506 + 2.8 %	0.496 + 0.8 %	0.560 + 13.7 %	**0.640** + 29.9 %	0.622 + 26.3 %
10	0.570	0.586 + 2.8 %	0.571 + 0.2 %	0.630 + 10.5 %	**0.718** + 26.0 %	0.717 + 25.9 %
15	0.610	0.630 + 3.2 %	0.636 + 4.2 %	0.687 + 12.7 %	**0.798** + 30.8 %	0.776 + 27.1 %
20	0.691	0.688 -0.4 %	0.697 + 0.9 %	0.745 + 7.8 %	**0.835** + 20.9 %	**0.835** + 20.9 %
25	0.738	0.749 + 1.6 %	0.729 - 1.2 %	0.790 + 7.0 %	0.860 + 16.6 %	**0.868** + 17.7 %
30	0.769	0.778 + 1.1 %	0.763 + -0.8 %	0.817 + 6.3 %	0.886 + 15.2 %	**0.889** + 15.6 %
MAP	0.645	0.656 + 1.7 %	0.648 + 0.6 %	0.705 + 9.3 %	**0.789** + 22.4 %	0.784 + 21.6 %

Table V. The *Average Precision* performance of the top 20 returned results with different **Batch Sizes** on the COREL image testbed.

4.6 Experiment III: Varied Batch Size

The third set of experiments is to evaluate the performance with varied batch size. Table V and Table VI show the average precision performance on the top 20 returned results with three active learning iterations on both datasets by varying the batch size and fixing the label size to 10. Similar to the previous subsection, we omit the results for $\mathrm{SVM_{BAML}^{(CO)}}$ and $\mathrm{SVM_{BMA}^{(QP)}}$.

Similar to previous observations, the two proposed algorithms $\mathrm{SVM_{BMAL}^{SS(QP)}}$ and $\mathrm{SVM_{BMAL}^{SS(CO)}}$ consistently outperform the other four approaches with significant improvements. By ex-

Batch Size	$\mathrm{SVM_{AL}}$	$\mathrm{SVM_{AL}^{DIVA}}$	$\mathrm{SVM_{AL}^{DIVE}}$	SSAL	$\mathrm{SVM_{BMAL}^{SS(QP)}}$	$\mathrm{SVM_{BMAL}^{SS(CO)}}$
5	0.595	0.592 - 0.5 %	0.590 - 0.9 %	0.653 + 9.7 %	0.689 + 15.8 %	**0.697** + 17.1 %
10	0.674	0.701 + 3.9 %	0.698 + 3.5 %	0.748 + 10.9 %	**0.802** + 18.9 %	0.797 + 18.2 %
15	0.755	0.759 + 0.6 %	0.757 + 0.3 %	0.803 + 6.4 %	0.850 + 12.6 %	**0.852** + 12.8 %
20	0.793	0.808 + 1.9 %	0.807 + 1.8 %	0.835 + 5.4 %	0.878 + 10.7 %	**0.882** + 11.3 %
25	0.832	0.848 + 2.0 %	0.850 + 2.2 %	0.862 + 3.6 %	**0.902** + 8.5 %	0.900 + 8.3 %
30	0.852	0.868 + 1.8 %	0.874 + 2.6 %	0.875 + 2.6 %	**0.918** + 7.7 %	0.913 + 7.1 %
MAP	0.750	0.763 + 1.7 %	0.763 + 1.7 %	0.796 + 6.1 %	0.840 + 11.9 %	**0.840** + 12.0 %

Table VI. The *Average Precision* performance of the top 20 returned results with different **Batch Sizes** on the ImageCLEF medical image testbed.

amining the results in detail, we found that when the batch size increases, the relative improvements achieved by our algorithms compared to SSAL tend to become more significant. For example, on the ImageCLEF dataset, when the batch size equals 10, the improvement of $\mathrm{SVM_{BMAL}^{SS(QP)}}$ over the baseline is about 1.6 times the improvement achieved by SSAL. This ratio increases to 3 when the batch size is increased to 30. Similar observations are also found in the average recall results for the top 100 returned images as shown in Table VII and Table VIII. These results again show that the proposed batch mode active learning method is more effective for selecting a batch of informative unlabeled examples for relevance feedback in CBIR.

Batch Size	$\mathrm{SVM_{AL}}$	$\mathrm{SVM_{AL}^{DIVA}}$	$\mathrm{SVM_{AL}^{DIVE}}$	SSAL	$\mathrm{SVM_{BMAL}^{SS(QP)}}$	$\mathrm{SVM_{BMAL}^{SS(CO)}}$
5	0.248	0.254 + 2.4 %	0.249 + 0.4 %	0.272 + 9.6 %	**0.332** + 33.9 %	0.321 + 29.4 %
10	0.291	0.301 + 3.4 %	0.289 -0.7 %	0.316 + 8.5 %	0.373 + 28.1 %	**0.377** + 29.5 %
15	0.317	0.325 + 2.7 %	0.327 + 3.4 %	0.349 + 10.1 %	**0.423** + 33.7 %	0.412 + 30.2 %
20	0.351	0.354 + 0.7 %	0.358 + 1.9 %	0.380 + 8.1 %	**0.451** + 28.3 %	0.447 + 27.2 %
25	0.376	0.380 + 1.1 %	0.371 -1.3 %	0.409 + 8.8 %	0.468 + 24.6 %	**0.471** + 25.4 %
30	0.393	0.401 + 2.2 %	0.398 + 1.3 %	0.427 + 8.6 %	0.490 + 24.8 %	**0.493** + 25.5 %
MAR	0.329	0.336 + 2.0 %	0.332 + 0.8 %	0.358 + 8.9 %	**0.423** + 28.4 %	0.420 + 27.6 %

Table VII. The *Average Recall* performance of the top 100 returned results with different **Batch Sizes** on the COREL image testbed.

Batch Size	$\mathrm{SVM_{AL}}$	$\mathrm{SVM_{AL}^{DIVA}}$	$\mathrm{SVM_{AL}^{DIVE}}$	SSAL	$\mathrm{SVM_{BMAL}^{SS(QP)}}$	$\mathrm{SVM_{BMAL}^{SS(CO)}}$
5	0.113	0.116 + 2.9 %	0.116 + 2.3 %	0.119 + 5.2 %	**0.139** + 22.9 %	0.138 + 22.5 %
10	0.134	0.140 + 4.9 %	0.138 + 3.0 %	0.140 + 4.9 %	**0.166** + 24.1 %	0.164 + 22.7 %
15	0.153	0.153 + 0.2 %	0.154 + 0.9 %	0.156 + 2.3 %	**0.184** + 20.4 %	**0.184** + 20.5 %
20	0.162	0.166 + 2.7 %	0.169 + 4.6 %	0.168 + 3.7 %	0.198 + 22.2 %	**0.199** + 23.0 %
25	0.177	0.180 + 2.0 %	0.186 + 5.6 %	0.183 + 3.8 %	0.214 + 21.3 %	**0.217** + 22.9 %
30	0.182	0.192 + 5.5 %	0.197 + 8.1 %	0.191 + 4.8 %	**0.228** + 24.9 %	0.225 + 23.7 %
MAR	0.153	0.158 + 3.1 %	0.160 + 4.3 %	0.160 + 4.1 %	**0.188** + 22.6 %	**0.188** + 22.6 %

Table VIII. The *Average Recall* performance of the top 100 returned results with different **Batch Sizes** on the ImageCLEF medical image testbed.

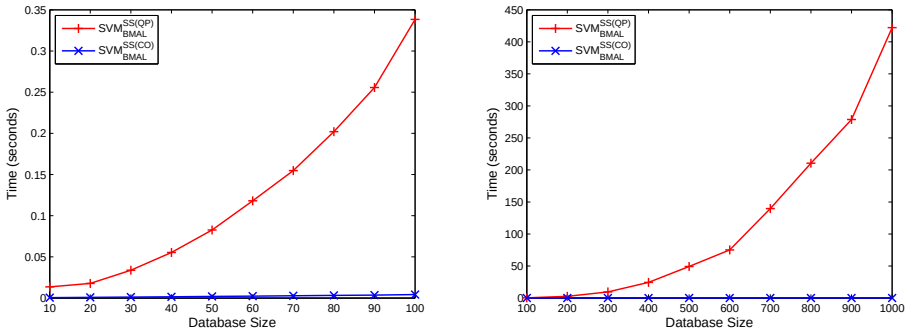
4.7 Experiment IV: Efficiency and Scalability of the Proposed Algorithms

The last experiment is to evaluate the efficiency and scalability performance of the two proposed algorithms: $\mathrm{SVM_{BMAL}^{SS(QP)}}$ and $\mathrm{SVM_{BMAL}^{SS(CO)}}$. To this purpose, we measure the time cost of the two algorithms with respect to different database sizes. Fig. 6 shows the results of average time performance of the two proposed algorithms for an active learning round with different database sizes where both the label size and the batch size are fixed to 10. Note that we do not count in the SVM training time, but focus on comparing the time used for the batch sampling task.

From the results, we clearly see that the combinatorial optimization approach with the greedy algorithm is significantly more efficient and scalable than the QP approach. As we observe, when the database size increases, the time cost of $\mathrm{SVM_{BMAL}^{SS(QP)}}$ increases dramatically, while the $\mathrm{SVM_{BMAL}^{SS(CO)}}$ increases linearly. Specifically, when the database size equals 1000, $\mathrm{SVM_{BMAL}^{SS(QP)}}$ takes about 420 seconds, while $\mathrm{SVM_{BMAL}^{SS(CO)}}$ needs only about 0.06 second. Hence, we can conclude that the $\mathrm{SVM_{BMAL}^{SS(CO)}}$ solution, with comparable retrieval performance, is more efficient and scalable than $\mathrm{SVM_{BMAL}^{SS(QP)}}$ for large applications. Finally, as indicated in Figure 6, the time cost of $\mathrm{SVM_{BMAL}^{SS(CO)}}$ is very small (less a millisecond even for selecting 100 examples), and is almost ignorable when compared to training a SVM classifier. As a result, the computational time of $\mathrm{SVM_{BMAL}^{SS(CO)}}$, a greedy implementation of semi-supervised batch mode active learning, is almost dictated by the training of SVM classifiers.

5. CONCLUSIONS

We proposed a novel semi-supervised SVM batch mode active learning scheme for solving relevance feedback in content-based image retrieval, which explicitly addressed two main drawbacks of the regular SVM active learning. In particular, we presented a unified learning framework incorporating both labeled and unlabeled data to improve the retrieval accuracy, and developed a new batch mode active learning scheme based on the min-max



(a) database size of [10,100] (b) database size of [100, 1000]

Fig. 6. Time performance of the two proposed algorithms.

framework. We proposed two novel algorithms to solve the batch mode active learning problem effectively and efficiently. We conducted an extensive set of experiments to evaluate the performance of our techniques for relevance feedback in CBIR, from which the promising results showed the advantages of the proposed solution compared to several state-of-the-art methods.

Despite promising results, the proposed technique still suffers from the following limitations. First, the proposed approach does not explicitly address the issue of imbalanced class distribution, which is one of the critical issues in relevance feedback. Second, theoretic questions need to be investigated regarding how the proposed active learning method affects the generalization error of classification models. In addition, we aim to further improve the efficacy of the proposed greedy algorithm. To this end, we plan to alleviate the greedy nature of the algorithm by exploring the backward and the forward method, which is employed in feature selection. More specifically, we first conduct the forward selection procedure by following the greedy algorithm presented in this paper. With the k unlabeled examples selected by the greedy algorithm, we will then conduct the backward refinement by trying to replace each selected unlabeled example with other unlabeled examples. We expect the backward refinement to further improved the quality of selected image examples, and therefore enhance the retrieval accuracy.

6. ACKNOWLEDGMENTS

The work was supported in part by the National Science Foundation (IIS-0643494), National Institute of Health (1R01-GM079688-01), Singapore MOE Academic Tier-1 Research Grant (RG67/07), the grant of 2008 Microsoft Research Asia's Mobile in Computing Education research theme, and the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4150/07E). 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 A: Proof of Theorem 1

PROOF. First, we have $g(f, \mathcal{L} \cup (\mathbf{x}, y), K)$ written as

$$g(f, \mathcal{L} \cup (\mathbf{x}, y), K) = \frac{\lambda}{2}||f||^2_{\mathcal{H}_K} + l(y, f(\mathbf{x})) + \sum_{i=1}^{l} l(y_i, f(\mathbf{x}_i))$$

Since

$$\max_{y \in \{-1,+1\}} l(y, f(\mathbf{x})) = \max_{y \in \{-1,+1\}} \max(0, 1 - yf(\mathbf{x})) = 1 + |f(\mathbf{x})|,$$

the problem in (7) can be rewritten as

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{U}} |f(\mathbf{x})| + \left[1 + \frac{\lambda}{2}||f||^2_{\mathcal{H}_K} + \sum_{i=1}^{l} l(y_i, f(\mathbf{x}_i))\right]$$

Since the second term is independent form $\mathbf{x}$, the above problem is equivalent to (6). □

Appendix B: Proof of Theorem 2

PROOF. First, note that

$$\begin{aligned}\max_{\mathbf{y} \in \{-1,+1\}^k} \min_{f \in \mathcal{H}_K} g(f, \mathcal{L} \cup (\mathcal{S}, \mathbf{y}), K) &= \max_{\mathbf{y} \in [-1,+1]^k} \min_{f \in \mathcal{H}_K} g(f, \mathcal{L} \cup (\mathcal{S}, \mathbf{y}), K) \\ &= \min_{f \in \mathcal{H}_K} \max_{\mathbf{y} \in [-1,+1]^k} g(f, \mathcal{L} \cup (\mathcal{S}, \mathbf{y}), K)\end{aligned}$$

In the last step, we apply the von Neuman lemma to switch **min** with **max** because $g(f, \mathcal{L} \cup (\mathcal{S}, \mathbf{y}), K)$ is concave in $\mathbf{y}$ and convex in $f(\cdot)$. We then examine quantity $\max_{\mathbf{y} \in [-1,+1]^k} g(f, \mathcal{L} \cup (\mathcal{S}, \mathbf{y}), K)$, which can be simplified as follows:

$$\begin{aligned}&\max_{\mathbf{y} \in \{-1,+1\}^k} g(f, \mathcal{L} \cup (\mathcal{S}, \mathbf{y}), K) \\ =\ & \frac{\lambda}{2}\|f\|^2_{\mathcal{H}_K} + \sum_{i=1}^{l} l(y_i, f(\mathbf{x}_i) + \sum_{\mathbf{x}_j \in \mathcal{S}} \max_{y_j \in \{-1,+1\}} l(y_j, f(\mathbf{x}_j)) \\ =\ & \frac{\lambda}{2}\|f\|^2_{\mathcal{H}_K} + \sum_{i=1}^{l} l(y_i, f(\mathbf{x}_i) + \sum_{\mathbf{x}_j \in \mathcal{S}} \max(0, 1 + f(\mathbf{x}_j), 1 - f(\mathbf{x}_j)) \\ =\ & \frac{\lambda}{2}\|f\|^2_{\mathcal{H}_K} + \sum_{i=1}^{l} l(y_i, f(\mathbf{x}_i) + \sum_{\mathbf{x}_j \in \mathcal{S}} (1 + |f(\mathbf{x}_j)|)\end{aligned}$$

By removing constant 1 from the above equation, we have the result in the theorem. □

Appendix C: Proof of Theorem 3

PROOF. First, note that the objective function in (12), i.e., $g(f, \mathcal{L}, \mathbf{q}, K)$, is linear in $\mathbf{q}$. Therefore, according to linear programming results, one of the optimal solutions to (12) should be its extreme point, which corresponds to a binary solution for $\mathbf{q}$. Hence, the optimal solution to (12) is indeed a feasible solution for (10). Second, since the optimal value for (12) is no larger than the optimal value for (10), the binary optimal solution $\mathbf{q}$ found by (12) is guaranteed to be an optimal solution for (10). We thus conclude the equivalence between (10) and (12). □

Appendix D: Proof of Theorem 4

PROOF. First, we derive the dual form of (12), i.e.,

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^l,\boldsymbol{\gamma}\in\mathbb{R}^{n-l}} \sum_{i=1}^{l}\alpha_i + \sum_{j=1}^{n-l}|\gamma_j| - \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})^\top K_{l,l}(\boldsymbol{\alpha}\circ\boldsymbol{y})$$
$$-\frac{1}{2}\boldsymbol{\gamma}^\top K_{u,u}\boldsymbol{\gamma} - (\boldsymbol{\alpha}\circ y)^\top K_{l,u}\boldsymbol{\gamma} \tag{25}$$
$$\text{s. t.} \qquad |\gamma_j| \le \frac{q_j}{\lambda}, j = 1,\ldots,n-l \tag{26}$$
$$0 \le \alpha_i \le \frac{1}{\lambda}, i = 1,\ldots,l \tag{27}$$

In above, the sub-indices l and u are used to refer to the columns and rows in matrix K that are related to labeled examples and unlabeled examples, respectively; operator $\circ$ stands for the element-wise product between two vectors.

We then rewrite the objective function in the dual in (25) into three parts, i.e.,

$$h_{l,l} = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})^\top K_{l,l}(\boldsymbol{\alpha}\circ\mathbf{y})$$
$$h_{u,u} = \sum_{j=1}^{n-l}|\gamma_j| - \frac{1}{2}\gamma^\top K_{u,u}\gamma \qquad h_{l,u} = (\boldsymbol{\alpha}\circ\mathbf{y})^\top K_{l,u}\gamma$$

Then, the optimal value for $\min_{f\in\mathcal{H}_K}\tilde{g}(f,\mathcal{L},\mathbf{q},K)$ is upper bounded by $\tilde{h}_{u,u}+\tilde{h}_{l,l}+\tilde{h}_{l,u}$ where $\tilde{h}_{u,u}$, $\tilde{h}_{l,l}$, and $\tilde{h}_{l,u}$ are defined as follows:

$$\tilde{h}_{l,l} = \max_{0\preceq\boldsymbol{\alpha}\preceq 1/\lambda} h_{l,l}, \quad \tilde{h}_{u,u} = \max_{|\boldsymbol{\gamma}|\preceq\mathbf{q}/\lambda} h_{u,u}, \quad \tilde{h}_{l,u} = \max_{0\preceq\boldsymbol{\alpha}\preceq 1/\lambda,|\boldsymbol{\gamma}|\preceq\mathbf{q}/\lambda} h_{l,u}$$

Note since $\max_{0\preceq\boldsymbol{\alpha}\preceq 1/\lambda} h_{l,l} = \min_{f\in\mathcal{H}_K} g(f,\mathcal{L},K)$, we have $\tilde{h}_{l,l} = g(f^*,\mathcal{L},K)$. Furthermore, we can bound $\tilde{h}_{u,u}$ as follows:

$$\tilde{h}_{u,u} \le \frac{k}{\lambda} - \min_{|\boldsymbol{\gamma}|\le\mathbf{q}/\lambda}\boldsymbol{\gamma}^\top K_{u,u}\boldsymbol{\gamma} \le \frac{k}{\lambda} + \frac{1}{2}\mathbf{q}^\top K_{u,u}\mathbf{q}.$$

Finally, $\tilde{h}_{l,u}$ is bounded by

$$\tilde{h}_{l,u} \le -[\boldsymbol{\alpha}^*]^\top K_{l,u}\boldsymbol{\gamma} = -\sum_{j=1}^{n-l}\gamma_j\sum_{i=1}^{l}\alpha_i^* y_i k(\mathbf{x}_{j+l},\mathbf{x}_i) \le \frac{1}{\lambda}\mathbf{q}^\top\tilde{\mathbf{f}}$$

where $\boldsymbol{\alpha}^*$ are the optimal solution to the dual problem of $\min_{f\in\mathcal{H}_K} g(f,\mathcal{L},K)$. Combining the above three bounds together, we have the result in Theorem 4. □

Appendix E: Proof of Theorem 6

PROOF. As we already pointed out, the key is to show that $g(\mathcal{S})$ defined in (19) satisfies the three conditions specified in Theorem 5. First, we show $g(\mathcal{S})$ is a non-decreasing set function. Without loss of generality, we consider set $\mathcal{A}$ and $\mathcal{B} = \mathcal{A}\cup i$ while $i\notin\mathcal{A}$. It is

sufficient to show

$$g(\mathcal{B}) \geq g(\mathcal{A})$$

for any set $\mathcal{A}$ and any element $i \notin \mathcal{A}$. We thus compute the difference $g(\mathcal{B}) - g(\mathcal{A})$, i.e.,

$$g(\mathcal{B}) - g(\mathcal{A}) \;=\; \tilde{f}_0 - f_i + \frac{\lambda}{2}\left(\theta\sqrt{n} - 2\sum_{j\in\mathcal{A}}[K_{u,u}]_{i,j}\right)$$

It is clear that the first two terms are non-negative since $\tilde{f}_0$ is the maximum value among all the unlabeled examples. We now show the term $\theta\sqrt{n} - 2\sum_{j\in\mathcal{A}}[K_{u,u}]_{i,j}$ is also non-negative. To this end, we consider the submatrix

$$\begin{pmatrix} K_{u,u}^{\mathcal{A}} & \mathbf{k}_i^{\mathcal{A}} \\ [\mathbf{k}_i^{\mathcal{A}}]^\top & [K_{u,u}]_{i,i} \end{pmatrix}$$

where $K_{u,u}^{\mathcal{A}}$ refers to the submatrix of $K_{u,u}$ that involves the examples in $\mathcal{A}$. $\mathbf{k}_i^{\mathcal{A}}$ includes the kernel similarity between the ith example and the examples in $\mathcal{A}$. Since $K_{u,u} \succeq 0$, according to the Schur complement, we have

$$\begin{aligned}[K_{u,u}]_{i,i} \;&\geq\; [\mathbf{k}_i^{\mathcal{A}}]^\top [K_{u,u}^{\mathcal{A}}]^{-1}\mathbf{k}_i^{\mathcal{A}} \geq \frac{1}{\theta - [K_{u,u}]_{i,i}}\|\mathbf{k}_i^{\mathcal{A}}\|_2^2 \\ &\geq\; \frac{1}{(\theta - [K_{u,u}]_{i,i})|\mathcal{A}|}\left(\sum_{j\in\mathcal{A}}[K_{u,u}]_{i,j}\right)^2\end{aligned}$$

In the above derivation, the second inequality follows the fact

$$K_{u,u}^{\mathcal{A}} \leq \mathrm{tr}(K_{u,u}^{\mathcal{A}})I \leq (\theta - [K_{u,u}]_{i,i})I,$$

and the last inequality uses the Cauchy inequality. Using the above result, we have

$$\sum_{j\in\mathcal{A}}[K_{u,u}]_{i,j} \leq \sqrt{[K_{u,u}]_{i,i}(\theta - [K_{u,u}]_{i,i})|\mathcal{A}|} \leq \frac{\sqrt{n}}{2}\theta,$$

and therefore have $g(\mathcal{A}) \leq g(\mathcal{B})$ when $\mathcal{B} = \mathcal{A} \cup i$.

The third property, i.e., $g(\emptyset) = 0$, can be easily verified. We thus focus on proving the second property, i.e., $g(\mathcal{S})$ is a submodular function. It is sufficient to show for any set $\mathcal{A}$, and two elements i and j that do not belong to $\mathcal{A}$, we have

$$g(\mathcal{A} \cup j) - g(\mathcal{A}) \geq g(\mathcal{A} \cup \{i, j\}) - g(\mathcal{A} \cup i)$$

To this end, we evaluate the quantity $g(\mathcal{A} \cup j) - g(\mathcal{A}) - g(\mathcal{A} \cup \{i, j\}) + g(\mathcal{A} \cup i)$, which results in the following expression:

$$g(\mathcal{A} \cup j) - g(\mathcal{A}) - g(\mathcal{A} \cup \{i, j\}) + g(\mathcal{A} \cup i) = \lambda[K_{u,u}]_{i,j} \geq 0$$

Therefore, $g(\mathcal{A})$ is a submodular function. Using the result in Theorem 5, we prove Theorem 6. □

REFERENCES

BOYD, S. AND VANDENBERGHE, L. 2004. *Convex Optimization*. Cambridge University Press.

BRINKER, K. 2003. Incorporating diversity in active learning with support vector machines. In *Proc. ICML2003*.

CHAPELLE, O., SCHÖLKOPF, B., AND ZIEN, A. 2006. *Semi-Supervised Learning*. The MIT Press.

COHN, D. A., GHAHRAMANI, Z., AND JORDAN, M. I. 1995. Active learning with statistical models. In *Proc. NIPS*.

DAGLI, C. K., RAJARAM, S., AND HUANG, T. S. 2006. Leveraging active learning for relevance feedback using an information theoretic diversity measure. In *ACM Conferece on Image and Video Retrieval (CIVR), Lecture Notes in Computer Science*. 123–132.

GOH, K.-S., CHANG, E. Y., AND LAI, W.-C. 2004. Multimodal concept-dependent active learning for image retrieval. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*. ACM, New York, NY, USA, 564–571.

HOI, S. C., JIN, R., ZHU, J., AND LYU, M. R. 2006. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd International Conference on Machine Learning (ICML2006)*. Pittsburgh, PA, US.

HOI, S. C. H., JIN, R., AND LYU, M. R. 2006. Large-scale text categorization by batch mode active learning. In *Proc. WWW2006*. Edinburgh, England, UK.

HOI, S. C. H. AND LYU, M. R. 2005. A semi-supervised active learning framework for image retrieval. In *Proc. CVPR2005*.

HOI, S. C. H., LYU, M. R., AND CHANG, E. Y. 2006. Learning the unified kernel machines for classification. In *Proc. KDD 2006*.

HOI, S. C. H., LYU, M. R., AND JIN, R. 2006. A unified log-based relevance feedback scheme for image retrieval. *IEEE Transactions on Knowledge and Data Engineering (TKDE) 18,* 4, 509–204.

HONG, P., TIAN, Q., AND HUANG, T. S. 2000. Incorporate support vector machines to content-based image retrieval with relevant feedback. In *Proceedings of IEEE International Conference on Image Processing (ICIP 2000)*. Vol. 3. Vancouver, BC, Canada, 750–753.

HUANG, T. S. AND ZHOU, X. S. 2001. Image retrieval by relevance feedback: from heuristic weight adjustment to optimal learning methods. In *Proceedings of the IEEE International Conference on Image Processing (ICIP 2001)*. Vol. 3. Thessaloniki, Greece, 2–5.

ISHIKAWA, Y., SUBRAMANYA, R., AND FALOUTSOS, C. 1998. MindReader: Querying databases through multiple examples. In *Proceedings of the 24th International Conference on Very Large Data Bases (VLDB 1998)*. 218–227.

LAAKSONEN, J., KOSKELA, M., AND OJA, E. 1999. Picsom: Self-organizing maps for content-based image retrieval. In *Proc. International Joint Conference on Neural Networks (IJCNN'99)*. Washington, DC, USA.

LEW, M. S., SEBE, N., DJERABA, C., AND JAIN, R. 2006. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl. 2,* 1, 1–19.

LI, J., ALLINSON, N., TAO, D., AND LI, X. 2006. Multitraining support vector machine for image retrieval. *IEEE Transactions on Image Processing 15,* 11, 3597–3601.

LIERE, R. AND TADEPALLI, P. 1997. Active learning with committees for text categorization. In *Proc. AAAI*.

MACARTHUR, S., BRODLEY, C., AND SHYU, C. 2000. Relevance feedback decision trees in content-based image retrieval. In *Proc. IEEE Workshop on Content-based Access of lmage and Video Libraries*. 68–72.

MANJUNATH, B. S. AND MA, W. Y. 1996. Texture features for browsing and retrieval of image data. *IEEE Trans. PAMI 18,* 8, 837–842.

MCCALLUM, A. K. AND NIGAM, K. 1998. Employing EM and pool-based active learning for text classification. In *Proc. ICML'98*.

MULLER, H., DESELAERS, T., LEHMANN, T., CLOUGH, P., AND HERSH, W. 2007. Overview of the imageclefmed 2006 medical retrieval annotation tasks. In *7th Workshop of Cross-Language Evaluation Forum (CLEF2006)*.

NEMHAUSER, G., WOLSEY, L., AND FISHER, M. 1978. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming 14*, 265–294.

PANDA, N., GOH, K.-S., AND CHANG, E. Y. 2006. Active learning in very large databases. *Journal of Multimedia Tools and Applications Special Issue on Computer Vision Meets Databases 31,* 3, 249–267.

QIAN, F., LI, M., ZHANG, L., ZHANG, H.-J., AND ZHANG, B. 2002. Gaussian mixture model for relevance feedback in image retrieval. In *Proceedings of International Conference on Multimedia and Expo (ICME'02)*. Vol. 1. 229–232.

RUI, Y., HUANG, T., AND MEHROTRA, S. 1997. Content-based image retrieval with relevance feedback in mars. II: 815–818.

RUI, Y., HUANG, T. S., ORTEGA, M., AND MEHROTRA, S. 1998. Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Trans. CSVT 8,* 5 (Sept.), 644–655.

SCHOHN, G. AND COHN, D. 2000. Less is more: Active learning with support vector machines. In *Proc. 17th ICML*.

SINDHWANI, V., NIYOGI, P., AND BELKIN, M. 2005. Beyond the point cloud: from transductive to semi-supervised learning. In *Proc. ICML 2005*.

SMEULDERS, A. W. M., WORRING, M., SANTINI, S., GUPTA, A., AND JAIN, R. 2000. Content-based image retrieval at the end of the early years. *IEEE Trans. PAMI 22,* 12, 1349–1380.

TAO, D., LI, X., AND MAYBANK, S. 2007. Negative samples analysis in relevance feedback. *IEEE Transactions on Image Processing 19,* 4, 568–580.

TAO, D., LI, X., AND MAYBANK, S. 2008. Which components are important for interactive image searching? *IEEE Transactions on Circuits and Systems for Video Technology 18,* 1, 1–11.

TAO, D. AND TANG, X. 2004a. Nonparametric discriminant analysis in relevance feedback for content-based image retrieval. In *IEEE International Conference on Pattern Recognition (ICPR'04)*. 1013–1016.

TAO, D. AND TANG, X. 2004b. Random sampling based svm for relevance feedback image retrieval. In *Proceedings IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

TAO, D., TANG, X., LI, X., AND RUI, Y. 2006. Direct kernel biased discriminant analysis: a new content-based image retrieval relevance feedback algorithm. *IEEE Transactions on Image Processing 8,* 4, 716–727.

TIEU, K. AND VIOLA, P. 2000. Boosting image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000)*. Vol. 1. South Carolina, USA, 228–235.

TONG, S. AND CHANG, E. 2001. Support vector machine active learning for image retrieval. In *Proc. ACM Multimedia Conference*.

TONG, S. AND KOLLER, D. 2000. Support vector machine active learning with applications to text classification. In *Proc. 17th ICML*.

VAPNIK, V. N. 1998. *Statistical Learning Theory*. Wiley.

VASCONCELOS, N. AND LIPPMAN, A. 1999. Learning from user feedback in image retrieval systems. In *Advances in Neural Information Processing Systems*.

WANG, L., CHAN, K. L., AND ZHANG, Z. 2003. Bootstrapping SVM active learning by incorporating unlabelled images for image retrieval. In *Proc. CVPR*.

WU, Y., TIAN, Q., AND HUANG, T. S. 2000. Discriminant-em algorithm with application to image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00)*. South Carolina, USA.

YUHONG GUO, D. S. 2007. Discriminative batch mode active learning. In *Proceedings of Advances in Neural Information Processing Systems (NIPS2007)*.

ZHANG, L., LIN, F., AND ZHANG, B. 2001. Support vector machine learning for image retrieval. In *Proceedings of the International Conference on Image Processing (ICIP 2001)*. Vol. 2. 721–724.

ZHANG, T. AND ANDO, R. K. 2005. Analysis of spectral kernel design based semi-supervised learning. In *NIPS*.

ZHOU, X. S. AND HUANG, T. S. 2001. Small sample learning during multimedia retrieval using biasmap. In *Proceedings IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

ZHOU, X. S. AND HUANG, T. S. 2003. Relevance feedback for image retrieval: a comprehensive review. *ACM Multimedia Systems 8,* 6, 536–544.

ZHOU, Z.-H., CHEN, K.-J., AND DAI, H.-B. 2006. Enhancing relevance feedback in image retrieval using unlabeled data. *ACM Trans. Inf. Syst. 24,* 2, 219–244.