Semi-Supervised SVM Batch Mode Active Learning for Image Retrieval

Steven C.H. Hoi
School of Computer Engineering
Nanyang Technological University
chhoi@ntu.edu.sg

Rong Jin
Dept. of Computer Sci. & Eng.
Michigan State University
rongjin@cse.msu.edu

Jianke Zhu
Dept. of Computer Sci. & Eng.
Chinese University of Hong Kong
jkzhu@cse.cuhk.edu.hk

Michael R. Lyu
Dept. of Computer Sci. & Eng.
Chinese University of Hong Kong
lyu@cse.cuhk.edu.hk

Abstract

Active learning has been shown as a key technique for improving content-based image retrieval (CBIR) performance. Among various methods, support vector machine (SVM) active learning is popular for its application to relevance feedback in CBIR. However, the regular SVM active learning has two main drawbacks when used for relevance feedback. First, SVM often suffers from learning with a small number of labeled examples, which is the case in relevance feedback. Second, SVM active learning usually does not take into account the redundancy among examples, and therefore could select multiple examples in relevance feedback that are similar (or even identical) to each other. In this paper, we propose a novel scheme that exploits both semi-supervised kernel learning and batch mode active learning for relevance feedback in CBIR. In particular, a kernel function is first learned from a mixture of labeled and unlabeled examples. The kernel will then be used to effectively identify the informative and diverse examples for active learning via a min-max framework. An empirical study with relevance feedback of CBIR showed that the proposed scheme is significantly more effective than other state-of-the-art approaches.

1. Introduction

Learning with user's interactions is crucial to many applications in computer vision and pattern recognition. One of them is content-based image retrieval (CBIR) where users are often engaged to interact with the CBIR system for improving the retrieval quality [16]. Such an interactive procedure is often known as *relevance feedback* [13], where the CBIR system attempts to understand the user's information needs by learning from the feedback examples judged by users. Due to the challenge of the semantic gap, traditional relevance feedback techniques often have to repeat many runs in order to achieve desirable results. To reduce the number of labeled examples required by relevance feedback, one key issue is how to identify the most informative unlabeled examples such that the retrieval performance could be improved most efficiently. Active learning is an important technique to address this challenge.

Unlike relevance feedback with passive learning where retrieved examples are presented according to their relevance to a given query, active learning for relevance feedback aims at identifying and presenting users with the examples that are deemed informative regarding their searching needs. A popular active learning technique in CBIR is called support vector machine (SVM) active learning [17], which learns an SVM model from the user feedback examples and employs it to find the most informative unlabeled examples. SVM active learning has been shown to outperform relevance feedback with passive learning [17]. However, SVM active learning has two main shortcomings when deployed to relevance feedback in CBIR.

First, SVM may fail to learn an accurate classification model from a small number of labeled examples. Given the limited number of labeled examples that are collected in relevance feedback of CBIR, directly applying an SVM model for active learning may not improve the retrieval accuracy significantly. Second, most relevance feedback solutions for CBIR present users with a number of retrieved examples while the SVM active learning method is designed to select a single example for each learning iteration. As a consequence, directly applying the SVM active learning method to relevance feedback may result in the selection of multiple examples that are similar (or even identical) to each other. We refer to these two problems as "*small training size problem*" and "*batch sampling problem*", respectively.

To address the above problems, we propose a novel

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

scheme of Semi-Supervised Support Vector Machine Batch Mode Active Learning, $\text{SVM}^{\text{SS}}_{\text{BMAL}}$ for short. It handles the small training size problem by a semi-supervised learning technique, and the batch sampling problem in active learning via a min-max framework. Our 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 presents the problem formulation and our solution. Section 3 gives empirical evaluations in CBIR. Section 4 discusses related work. Section 5 concludes this work.

2. Semi-supervised SVM Batch Mode Active Learning

We will first formulate relevance feedback as a problem of batch mode active learning, followed by the presentation of a semi-supervised kernel learning and the min-max framework for SVM batch mode active learning.

2.1. Preliminaries

In this paper, we focus on the problem of relevance feedback task in CBIR. Let's denote by $\mathcal{L} = \{(\mathbf{x}_1, y_i) \ldots, (\mathbf{x}_l, y_l)\}$ a set of l initially labeled image examples, 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. The learning problem in the relevance feedback of a CBIR retrieval task is how to identify the k unlabeled image examples, denoted by S^*, which can be used to improve the retrieval accuracy most efficiently. It can be formulated as the following combinatorial optimization problem:

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

where $risk(f, \mathcal{S}, \mathcal{L}, \mathcal{U})$ is 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. Since the above formulation selects multiple examples, we refer to the problem as "*batch mode active learning*". In general, finding an optimal solution of the above combinational optimization is an NP-hard problem. Regular active learning techniques only select a single unlabeled example (i.e., $k = 1$), and therefore avoid solving the combinatorial optimization problem.

Since our study is focused on applying SVM for batch mode active learning, here we first briefly review SVM. The key idea of SVM is to learn an optimal hyperplane that separates the training examples with the maximal margin [19]. A linear SVM finds an optimal hyperplane $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$ by solving the following optimization problem:

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

where λ is the regularization parameter and ξ_is are slack variables that are introduced for the nonseparable examples. To obtain a nonlinear decision boundary, kernel tricks are usually applied, in which the examples are projected from the original data space to a higher dimensional feature space where an optimal linear hyperplane is found to separate the data from different classes. It is also often formulated in a regularization learning framework as follows:

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

where f is the hyperplane function and can be written as $f(\mathbf{x}) = \sum_{i \in \mathcal{L}} \alpha_i K(\mathbf{x}, \mathbf{x}_i)$, and $\mathcal{H}_K$ is the Hilbert space reproduced by a kernel function K.

2.2. A Semi-supervised Support Vector Machine

Traditional SVM active learning often adopts a supervised learning solution to train a classifier $f(\cdot)$ on the labeled examples [17, 18]. The resulting decision function f^* may not be accurate when the number of labeled examples is limited. 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, and a variety of semi-supervised learning (SSL) techniques have been proposed [3]. In this paper, we employ a unified kernel learning approach by fusing both unsupervised kernel learning and supervised kernel classifier [10, 21]. The main idea is to first learn a data-dependent kernel from the unlabeled data, and then apply the learned kernel to train a supervised SVM classifier $f(\cdot)$ based on the regularization learning framework in (2). Compared with 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. In this paper, we adopt the kernel deformation principle for learning a data-dependent kernel from unlabeled data [15].

The main idea of kernel deformation is to first estimate the geometry of the underlying marginal distribution from both labeled and unlabeled data, and then derive a data-dependent kernel by incorporating the estimated geometry [15]. As a result, the derived kernel function is able to take advantage of the unlabeled data via the geometry.

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. In [15], the authors assume the following relationship between the two Hilbert spaces, i.e.,

$$\langle f, g \rangle_{\tilde{\mathcal{H}}} = \langle f, g \rangle_{\mathcal{H}} + \mathbf{f}^\top M \mathbf{g} \qquad (3)$$

where $f(\cdot)$ and $g(\cdot)$ are two functions, $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))$ evaluates the function $f(\cdot)$ 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 (3), 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. With the above assumption in (3), [15] derived the new kernel function $\tilde{k}(\cdot,\cdot)$ associated with the deformed space $\tilde{\mathcal{H}}$, i.e.,

$$\tilde{k}(\mathbf{x},\mathbf{y}) = k(\mathbf{x},\mathbf{y}) - \boldsymbol{\kappa}_{\mathbf{y}}^\top (I + MK)^{-1} M\boldsymbol{\kappa}_{\mathbf{x}} \qquad (4)$$

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 $(k(\mathbf{x}_1,\mathbf{x})\ldots k(\mathbf{x}_n,\mathbf{x}))^\top$. To capture the geometrical structure of data, a common approach is to define M as a function of graph Laplacian L, for example, $M = L^p$ where p is an integer. A graph Laplacian 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 $\exp(-|\mathbf{x}_i - \mathbf{x}_j|_2^2/\sigma^2)$. $\mathbf{1}$ is a vector of every element being 1. In our experiments, we set $p = 1$, i.e., $M = L$.

Remark. To better understand the kernel deformation, we can rewrite (4) 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.

2.3. SVM Batch Mode Active Learning

The traditional SVM active learning method employs the notion of *version space* for measuring the risk in the active learning task. Given the training data $\mathcal{L}$ and a Mercer kernel K, the version space is defined as a set of hyperplanes that can separate the training data in the feature space $\mathcal{H}_K$ induced by the Mercer kernel. More formally, the version space can be expressed as

$$\mathcal{V} = \{f \in \mathcal{H}_K | \forall i \in \{1,\ldots,l\}, \quad y_i f(\mathbf{x}_i) > 0\}.$$

The idea of SVM active learning is to find an optimal unlabeled example that will result in the maximal reduction of the version space. More details can be found in [18]. 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 approach to batch mode active learning.

2.3.1 Min-max View for Active Learning

Let $g(f,\mathcal{L},K)$ denote the margin-based objective function in the regularization framework in (2), i.e.,

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

To identify the most informative example, we adopt the worst case analysis by selecting the unlabeled example $\mathbf{x}$ that leads to a small value for the objective function $g(f,\mathcal{L},K)$ regardless of its assigned class label y. We can cast this idea into the following min-max framework:

$$\arg\min_{\mathbf{x}\in\mathcal{U}} \max_{y\in\{-1,+1\}} g(f, \mathcal{L}\cup(\mathbf{x},y), K) \qquad (5)$$

which can be further written explicitly as:

$$\arg\min_{\mathbf{x}_j\in\mathcal{U}} \max_{y_j\in\{-1,+1\}} \min_{f\in\mathcal{H}_K} \Big(\frac{\lambda}{2}||f||_{\mathcal{H}_K}^2 + \sum_{i\in\mathcal{L}\cup\{j\}} \big(y_i, f(\mathbf{x}_i)\big)_+\Big)$$

where $(\cdot,\cdot)_+$ is the hinge loss function, i.e. $\big(y_i, f(\mathbf{x}_i)\big)_+ = \max(0, 1 - y_i f(\mathbf{x}_i))$. Let f^* denote the optimal decision function found in (2), we can simplify the above optimization as follows:

$$\begin{aligned}
&\arg\min_{\mathbf{x}_j\in\mathcal{U}} \max_{y_j\in\{-1,+1\}} g(f, \mathcal{L}\cup\{(\mathbf{x}_j,y_j)\}, K)\\
&\approx \arg\min_{\mathbf{x}_j\in\mathcal{U}} \max_{y_j\in\{-1,+1\}} \Big(\max(0, 1 - y_j f^*(\mathbf{x}_j))\Big)\\
&= \arg\min_{\mathbf{x}_j\in\mathcal{U}} \Big(\max\big(0, 1 - f^*(\mathbf{x}_j), 1 + f^*(\mathbf{x}_j)\big)\Big)\\
&= \arg\min_{\mathbf{x}_j\in\mathcal{U}} \Big(1 + \big|f^*(\mathbf{x}_j)\big|\Big) = \arg\min_{\mathbf{x}_j\in\mathcal{U}} \big|f^*(\mathbf{x}_j)\big| \qquad (6)
\end{aligned}$$

The above result shows that an approximation to the min-max framework is to select the unlabeled example closest to the decision boundary f^* that is trained on the current set of labeled examples. This result is similar to the result derived from the version space analysis in [18]. Next, we will apply the min-max framework to batch mode active learning.

2.3.2 Min-max Framework for Batch Mode Active Learning

To extend the min-max framework for batch mode active learning, we consider the following optimization problem:

$$\arg\min_{\mathcal{S}\subseteq\mathcal{U}\wedge|\mathcal{S}|=k} \max_{\mathbf{y}\in\{-1,+1\}^k} \min_{f\in\mathcal{H}_K} g(f, \mathcal{L}\cup(\mathcal{S},\mathbf{y}), K) \qquad (7)$$

where $(\mathcal{S}, \mathbf{y}) = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathcal{S}\}$. To simplify the above problem, we introduce the objective function $\tilde{g}(f, \mathcal{L}, \mathcal{S}, K)$ as follows:

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

The following theorem shows a simplified form for the optimization problem in (7):

Theorem 1 *The optimization problem in (7) is equivalent to the following problem*

$$\operatorname*{arg\,min}_{\mathcal{S} \subseteq \mathcal{U} \wedge |\mathcal{S}| = k} \min_{f \in \mathcal{H}_K} \tilde{g}(f, \mathcal{L}, \mathcal{S}, K) \tag{8}$$

Compared to (7), the key advantage of the above formulation is that it removes the dependence of $\mathbf{y}$ from the objective function. The above theorem can be directly verified by taking the maximum over $\mathbf{y}$.

We will further simplify the above combinatorial optimization problem by introducing a probability q_i for each unlabeled example in $\mathcal{U}$ to represent the likelihood of selecting the example. The following theorem shows a continuous version of the optimization problem in (8) by using the probability q_i:

Theorem 2 *The optimization problem in (8) is equivalent to the following optimization problem:*

$$\operatorname*{arg\,min}_{\mathbf{q}^\top \mathbf{1} = k, \mathbf{q} \succeq 0} \min_{f \in \mathcal{H}_K} \tilde{g}(f, \mathcal{L}, \mathbf{q}, K) \tag{9}$$

where

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

The above theorem can be verified by using the result of linear programming, i.e., the optimal solution of a linear programming problem will always be its extreme point(s). In our case, this implies that the solution for q_i will either be zero or one.

To further carry out the analysis, we derive the dual form for the optimization problem $\min_{f \in \mathcal{H}_K} \tilde{g}(f, \mathcal{L}, \mathbf{q}, K)$, i.e.,

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

where the sub-indices l and u are used to refer to the columns and rows in matrix K that are related to the labeled examples and the unlabeled examples, respectively. The operator $\circ$ stands for the element-wise product between two vectors. The above dual form can be derived by first calculating the Lagrangian and then setting the first derivatives of the Lagrangian with respect to the primal variables to be zero. We use the above dual formulation to construct the upper bound for $\min_{f \in \mathcal{H}_K} \tilde{g}(f, \mathcal{L}, \mathbf{q}, K)$, and the result is summarized by the following theorem:

Theorem 3

$$\begin{aligned} & \min_{f \in \mathcal{H}_K} \tilde{g}(f, \mathcal{L}, \mathbf{q}, K) - \frac{k}{\lambda} \\ \le \quad & g(f^*, \mathcal{L}, K) + \frac{1}{\lambda}\mathbf{q}^\top \tilde{\mathbf{f}} + \frac{1}{2\lambda^2}\mathbf{q}^\top K_{u,u} \mathbf{q} \end{aligned}$$

where $\tilde{\mathbf{f}} = (|f^*(\mathbf{x}_{l+1})|, \ldots, |f^*(\mathbf{x}_n)|)^\top$ *and the function* $f^*(\mathbf{x})$ *is defined as* $f^* = \operatorname*{arg\,min}_{f \in \mathcal{H}_K} g(f, \mathcal{L}, K)$.

The above theorem follows directly from the following two equalities:

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

$$\max_{\boldsymbol{\gamma} \in \mathbb{R}^{n-l}} \sum_{j=1}^{n-l} |\gamma_j| \le \sum_{j=1}^{n-l} \frac{q_j}{\lambda} = \frac{k}{\lambda}$$

$$\min_{f \in \mathcal{H}_K} g(f, \mathcal{L}, K) = \max_{0 \le \alpha_i \le 1/\lambda} \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\boldsymbol{\alpha}^\top K_{l,l} \boldsymbol{\alpha}$$

Using the above upper bound in Theorem 3, we can now find the optimal $\mathbf{q}$ by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{q} \in \mathbb{R}^{n-l}} \quad & \mathbf{q}^\top \tilde{\mathbf{f}} + \frac{\lambda}{2}\mathbf{q}^\top K_{u,u} \mathbf{q} \\ \text{s. t.} \quad & \mathbf{q}^\top \mathbf{1} = k, \mathbf{0} \le \mathbf{q} \le \mathbf{1} \end{aligned} \tag{10}$$

where λ is a parameter that balances the two terms.

The above optimization is a standard quadratic programming (QP) problem that can be solved effectively by existing convex optimization software packages [1]. Finally, given the estimated q_i, we will select the unlabeled examples with the largest probabilities q_i. Figure 1 summarizes the overall algorithm for semi-supervised SVM batch mode active learning ($\text{SVM}^{\text{SS}}_{\text{BMAL}}$), which consists of two 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 of batch mode active learning.

Remark. It is interesting to examine the meaning of the the two terms in the objective function in (10). 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 prefer to select the examples that are close to the decision boundary. The second term, i.e., $\mathbf{q}^\top K_{u,u}\mathbf{q}$, is related to the redundancy among the selected examples. By minimizing $\mathbf{q}^\top K_{u,u}\mathbf{q}$, the selected examples tend to share small similarity among themselves.

Algorithm 1 Semi-Supervised SVM Batch Mode Active Learning

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 cost */

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 (4);

/* *Start batch mode active learning procedure* (Online) */

1: Train an SVM classifier: $f^* = \text{SVM\_Train}(\mathcal{L}, \tilde{\mathbf{K}})$;

2: Compute $\tilde{\mathbf{f}} = (|f^*(\mathbf{x}_{l+1})|, \ldots, |f^*(\mathbf{x}_n)|)^\top$

3: Find $\mathbf{q}$ by solving the QP problem in (10):

4: $\mathcal{S} = \varnothing$;

5: **while** ($|\mathcal{S}| < k$) **do**

6: $\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{U}} q(\mathbf{x})$;

7: $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathbf{x}^*\}$; $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\mathbf{x}^*\}$;

8: **end while**

9: **return** $\mathcal{S}$.

END

Figure 1. The proposed Semi-Supervised SVM Batch Mode Active Learning ($\text{SVM}^{\text{SS}}_{\text{BMAL}}$) algorithm

3. Experimental Results

To evaluate the performance of the proposed algorithm, we conduct a set of CBIR experiments by comparing it with several state-of-the-art active learning methods in image retrieval. Specifically, we design the experiments to evaluate two major factors that will affect the results of a batch mode active learning method. The first factor is the *label size*, i.e., the number of labeled images judged by a user in the first around of image retrieval in which no relevance feedback is applied. The second factor is the *batch size*, i.e., the number of data examples to be selected for labeling by active learning in each iteration of relevance feedback.

3.1. Experimental Testbed and Feature Extraction

We use the COREL photo images as our experimental testbed. In particular, we form a 20-category dataset that contains $2,000$ images from 20 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*, etc.

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 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 obtain the edges, from which the edge direction histogram is computed. The edge direction histogram is quantized into 18 bins of 20 degrees each, and 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 their entropy values. Thus, a 9-dimensional wavelet vector is employed.

3.2. Compared Schemes and Experimental Setup

In the experiments, we compare the proposed active learning algorithm ($\text{SVM}^{\text{SS}}_{\text{BMAL}}$) to the following algorithms for active learning:

- SVM Active Learning: the baseline method by the original SVM active learning algorithm that samples the examples closest to the decision boundary for labeling [17], denoted by SVM_{AL}.
- SVM Active Learning with Diversity: a heuristic modification of SVM active learning by incorporating diversity in the batch sampling procedure [2], denoted by $\text{SVM}^{\text{DIV}}_{\text{AL}}$.
- Semi-Supervised Active Learning: a fusion of semi-supervised learning and SVM active learning to overcome the small sample learning issue of regular SVM active learning [9], denoted by SSAL.

To evaluate the average performance, we conduct every experiment by a set of 200 random queries with images sampled from the datasets. We simulate the CBIR procedure by querying an image and returning the top images based on the Euclidean distances. The top l images are then labeled as the set of initially labeled data for relevance feedback. An RBF kernel with fixed kernel width is used for all the algorithms. Regarding the parameter setting, the penalty parameter C of SVM is set to 100 (or $\lambda = 0.01$) in all experiments, and the regularization parameter λ in the $\text{SVM}^{\text{SS}}_{\text{BMAL}}$ algorithm is simply set to 1 for all experiments. Finally, average precision (AP) and mean average precision (MAP) are adopted as the evaluation metric, in which the relevance judgements are based on whether the query image and the retrieved image belong to the same category [9, 17].

Label Size	$\mathrm{SVM_{AL}}$	$\mathrm{SVM_{AL}^{DIV}}$	SSAL	$\mathrm{SVM_{BMAL}^{SS}}$
5	0.361	0.370 + 2.4 %	0.399 + 10.5 %	0.435 + 20.4 %
10	0.401	0.409 + 2.0 %	0.449 + 11.9 %	0.486 + 21.1 %
15	0.441	0.447 + 1.3 %	0.487 + 10.3 %	0.539 + 22.2 %
20	0.463	0.464 + 0.4 %	0.511 + 10.4 %	0.556 + 20.1 %
25	0.496	0.499 + 0.7 %	0.537 + 8.4 %	0.581 + 17.3 %
30	0.522	0.524 + 0.3 %	0.566 + 8.3 %	0.601 + 15.1 %
MAP	0.447	0.452 + 1.1 %	0.491 + 9.9 %	0.533 + 19.2 %

Table 1. The *average precision* performance of top 50 returned results with different label sizes on the testbed.

3.3. Experiment I: Fixed Label and Batch Sizes

We first conduct experiments with both label size and batch size fixed to 10. Figure 2 shows the average precision for the first three rounds of relevance feedback on two datasets, respectively. In these figures, the blue and green dotted lines are for $\mathrm{SVM_{AL}}$ and $\mathrm{SVM_{AL}^{DIV}}$, respectively, and the pink and red solid lines are for SSAL and the proposed $\mathrm{SVM_{BMAL}^{SS}}$ algorithm, respectively.

Several observations can be drawn from the experimental results. First, for the first iteration as shown in Figure 2(a), we observe that both two semi-supervised learning solutions SSAL and $\mathrm{SVM_{BMAL}^{SS}}$ significantly outperform the other two supervised learning methods. Second, by examining the results with all three active learning iterations, we found that the heuristic $\mathrm{SVM_{AL}^{DIV}}$ method is only marginally better than the baseline method. In contrast, the two semi-supervised learning algorithms outperform the baseline method significantly for both active learning iterations. Finally, comparing the two semi-supervised learning algorithms, we found that the proposed $\mathrm{SVM_{BMAL}^{SS}}$ method achieves significantly better performance than SSAL.

3.4. Experiment II: Varied Label Size

Table 1 shows the experimental results of average precision for top 50 returned images with one active learning iteration for both datasets by varying the label size and fixing the batch size to 10. First, we observe that $\mathrm{SVM_{AL}^{DIV}}$ achieves no more than 3% improvement over the baseline. In contrast, SSAL achieves considerably better performance with 8% to 12% improvement over the baseline. Among all, the proposed algorithm achieves the best results, whose improvement almost doubles that of SSAL. In addition, we found that the average improvement with small label sizes is usually greater than that with large label sizes. For example, the relative improvement made by the proposed algorithm is 21.1% when label size is 10, and is reduced to 15.1% when the label size is 30. This again shows that the proposed algorithm is able to effectively address the problem of small training size.

3.5. Experiment III: Varied Batch Size

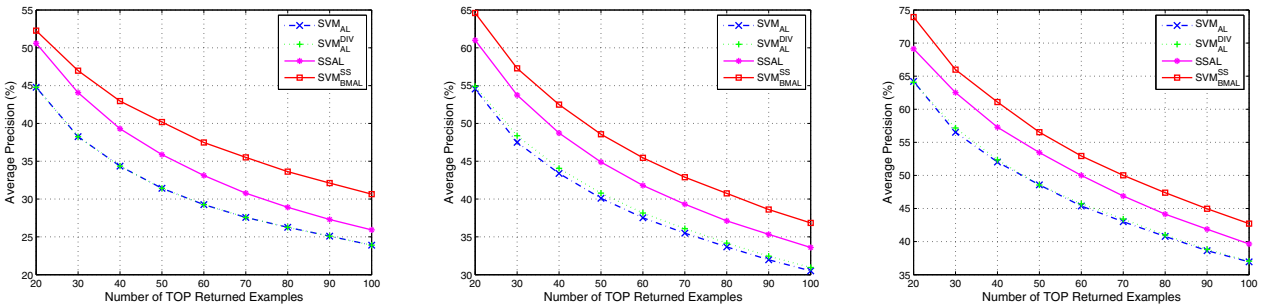
Table 2 shows the average precision performance on top 50 returned results with two active learning iterations on the two datasets by varying the batch size and fixing the label size to be 10. Similar to previous observations, the proposed algorithm consistently outperforms the other three approaches with significant improvement. By examining the results in detail, we found that when the batch size increases, the relative improvement by $\mathrm{SVM_{BMAL}^{SS}}$ tends to become more significant. For example, when the batch size equals to 5, the improvement of $\mathrm{SVM_{BMAL}^{SS}}$ over the baseline is about 1.6 times the improvement achieved by SSAL. This ratio increases to 2.3 when the batch size is increased to 25. These results again show that the proposed batch mode active learning algorithm is more effective in selecting a batch of informative unlabeled examples for labeling.

4. Related Work

Learning with relevance feedback in CBIR has been extensively studied, which has been shown as one way to attack the semantic gap issue [13, 16]. To improve the learning efficiency of relevance feedback, active learning has been studied in recent years. In machine learning, many active learning techniques have been proposed [4, 11, 12, 14, 18]. Due to limited space, we focus our main attention on those work in CBIR. A well-known and pioneering active learning work in CBIR is the SVM active learning proposed by Tong et al [17]. Its limitations have been addressed by some latter research efforts. For the small sample learning issue, Wang et al. proposed modifying the SVM active learning by engaging the unlabeled data with transductive SVM [20]. Hoi et al. developed a better solution to improve the limitation by combining semi-supervised learning techniques with SVM [9]. For the batch sampling issue, Brinker suggested a heuristic modification by incorporating diversity to select examples iteratively [2]. Some other different diversity measure method was also proposed [5]. Different from these heuristic approaches, we learn a sampling distribution by formally formulating the batch mode active learning problem. Finally, our work is different from some other recent work on batch mode active learning [8, 6, 7]. These studies were mainly based on kernel logistic regressions, which may not be able to applicable to SVM models directly.

5. Conclusion

We proposed a novel semi-supervised SVM batch mode active learning scheme for solving relevance feedback in content-based image retrieval that explicitly addressed the two main drawbacks of the well-known SVM active learning. In particular, we presented a unified learning framework of incorporating both labeled and unlabeled data



(a) 1st feedback round (b) 2nd feedback round (c) 3rd feedback round

Figure 2. The *average precision* performance of several active learning algorithms with fixed label and batch sizes on the testbed.

Batch Size	$\mathrm{SVM_{AL}}$	$\mathrm{SVM_{AL}^{DIV}}$	SSAL	$\mathrm{SVM_{BMAL}^{SS}}$
5	0.438	0.434 - 1.0 %	0.482 + 10.1 %	0.511 + 16.6 %
10	0.486	0.493 + 1.6 %	0.535 + 10.1 %	0.565 + 16.4 %
15	0.522	0.525 + 0.6 %	0.568 + 8.8 %	0.605 + 16.0 %
20	0.541	0.556 + 2.7 %	0.597 + 10.4 %	0.637 + 17.8 %
25	0.582	0.593 + 1.9 %	0.619 + 6.4 %	0.668 + 14.8 %
30	0.612	0.610 - 0.4 %	0.650 + 6.3 %	0.692 + 13.0 %
MAP	0.530	0.535 + 0.9 %	0.575 + 8.5 %	0.613 + 15.7 %

Table 2. The *average precision* performance of top 50 returned results with different batch sizes on the testbed.

to improve the retrieval accuracy, and developed a new batch mode active learning algorithm based on the min-max framework. The empirical results with relevance feedback of CBIR showed the advantages of the proposed solution compared to the other state-of-the-art methods. One limitation of our current solution is the QP algorithm, which may not be efficient for large-scale applications. In future work, we will study more efficient algorithms to address the efficiency and scalability issues.

Acknowledgments

The work was supported in part by the National Science Foundation (IIS-0643494), National Institute of Health (1R01-GM079688-01), Singapore NTU AcRF Tier-1 Research Grant (RG67/07), and Hong Kong RGC Grant (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.

References

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 4

[2] K. Brinker. Incorporating diversity in active learning with support vector machines. In *Proc. ICML2003*, 2003. 5, 6

[3] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. The MIT Press, 2006. 2

[4] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. In *Proc. NIPS*, 1995. 6

[5] C. K. Dagli, S. Rajaram, and T. S. Huang. 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*, pages 123–132, 2006. 6

[6] Y. Guo and D. Schuurmans. Discriminative batch mode active learning. In *Proceedings of Advances in Neural Information Processing Systems (NIPS2007)*, 2007. 6

[7] S. C. H. Hoi, R. Jin, and M. R. Lyu. Large-scale text categorization by batch mode active learning. In *Proc. WWW2006*, Edinburgh, England, UK, May 23–26 2006. 6

[8] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu. 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, June 25–29 2006. 6

[9] S. C. H. Hoi and M. R. Lyu. A semi-supervised active learning framework for image retrieval. In *Proc. CVPR2005*, 2005. 5, 6

[10] S. C. H. Hoi, M. R. Lyu, and E. Y. Chang. Learning the unified kernel machines for classification. In *Proc. KDD 2006*, 2006. 2

[11] R. Liere and P. Tadepalli. Active learning with committees for text categorization. In *Proc. AAAI*, 1997. 6

[12] A. K. McCallum and K. Nigam. Employing EM and pool-based active learning for text classification. In *Proc. ICML'98*, 1998. 6

[13] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Trans. CSVT*, 8(5):644–655, Sept. 1998. 1, 6

[14] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *Proc. 17th ICML*, 2000. 6

[15] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *Proc. ICML 2005*, 2005. 2, 3

[16] 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 Trans. PAMI*, 22(12):1349–1380, 2000. 1, 6

[17] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proc. ACM Multimedia Conference*, 2001. 1, 2, 5, 6

[18] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *Proc. 17th ICML*, 2000. 2, 3, 6

[19] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998. 2

[20] L. Wang, K. L. Chan, and Z. Zhang. Bootstrapping SVM active learning by incorporating unlabelled images for image retrieval. In *Proc. CVPR*, 2003. 6

[21] T. Zhang and R. K. Ando. Analysis of spectral kernel design based semi-supervised learning. In *NIPS*, 2005. 2