A Unified Log-Based Relevance Feedback Scheme for Image Retrieval

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

Abstract—Relevance feedback has emerged as a powerful tool to boost the retrieval performance in content-based image retrieval (CBIR). In the past, most research efforts in this field have focused on designing effective algorithms for traditional relevance feedback. Given that a CBIR system can collect and store users' relevance feedback information in a history log, an image retrieval system should be able to take advantage of the log data of users' feedback to enhance its retrieval performance. In this paper, we propose a unified framework for log-based relevance feedback that integrates the log of feedback data into the traditional relevance feedback schemes to learn effectively the correlation between low-level image features and high-level concepts. Given the error-prone nature of log data, we present a novel learning technique, named Soft Label Support Vector Machine, to tackle the noisy data problem. Extensive experiments are designed and conducted to evaluate the proposed algorithms based on the COREL image data set. The promising experimental results validate the effectiveness of our log-based relevance feedback scheme empirically.

Index Terms—Content-based image retrieval, relevance feedback, log-based relevance feedback, log data, user issues, semantic gap, support vector machines.

1 Introduction

1.1 Image Retrieval

With the rapid growth of digital devices for capturing and storing multimedia data, multimedia information retrieval has become one of the most important research topics in recent years, among which image retrieval has been one of the key challenging problems. In the image retrieval, content-based image retrieval (CBIR) is one of the most important topics which has attracted a broad range of research interests in many computer communities in the past decade [36]. Although extensive studies have been conducted, finding desired images from multimedia databases is still a challenging and open issue. The main challenges are due to two gaps in CBIR [36]. The first is the sensor gap between the object of the world and the information represented by computers. The second one is the semantic gap between the low-level visual features and high-level human perception and interpretation. Many early year studies on CBIR focused primarily on feature analysis which mainly aimed at solving the sensory gap [21], [37].

However, because of the complexity of image understanding and the challenge of semantic gap, it is impossible to discriminate all images by employing some rigid simple similarity measure on the low-level features. Although it is feasible to bridge the semantic gap by building an image index with textual descriptions, manual indexing on image databases is typically time-consuming, costly and subjective, and hence difficult to be fully deployed in practical applications. Despite the promising process recently reported in image annotations [3], [22], [25], fully automatic image annotation is still a long way off. Relevance feedback, as an alternative and more feasible technique to mitigate the semantic gap issue, has been intensively investigated in recent years [33].

1.2 Relevance Feedback

Relevance feedback originated from text-based information retrieval is a powerful technique to improve the retrieval performance [35]. In order to approach the query targets of an user, relevance feedback is viewed as the process of automatically altering an existing query by incorporating the relevance judgments that the user provide for the previously retrieved objectives. In image retrieval, relevance feedback will first solicit the user's relevance judgments on the retrieved images returned by CBIR systems. Then, it refines retrieval results by learning the query targets from the provided relevance information. Although relevance feedback was originated from text retrieval, it is a little bit surprising to see later on it attracted much more attentions in image retrieval. In the past decade, various relevance feedback techniques have been proposed, ranging from heuristic methods to many sophisticated learning techniques [7], [19], [44].

The early relevance feedback for image retrieval was typically inspired by traditional relevance feedback in text retrieval. For example, Rui et al. [33] proposed to learn on the ranks of the positive and negative images along the feature axis in the feature space, which is similar to the idea of learning on "term frequency" and "inverse term frequency" in text retrieval domain [32]. Later on, more systematic and comprehensive schemes were suggested to formulate the relevance feedback problem into an optimization problem. For example, MindReader formulated the feedback task as an optimization problem in which parameters are learned by minimizing the sum of overall distances from the query centroid to all relevant samples [20]. Rui et al. proposed a

- *S.C.H. Hoi and M.R. Lyu are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Ho Sin Hang Engineering Building, Shatin, N.T., Hong Kong. E-mail: {chhoi, lyu}@cse.cuhk.edu.hk.*
- *R. Jin is with the Department of Computer Science and Engineering, Michigan State University, 3115 Engineering Building, East Lansing, MI 48824. E-mail: rongjin@cse.msu.edu.*

Manuscript received 7 Mar. 2005; revised 27 July 2005; accepted 12 Oct. 2005; published online 17 Feb. 2006.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0081-0305.

more rigorous approach called "Optimizing Learning" that systematically formulates the relevance feedback as an optimizing problem and suggested a hierarchical learning approach rather than a flat model like the one from MindReader.

Recently, along with the rapid development in machine learning, a variety of machine learning techniques have been applied to the relevance feedback problem in image retrieval, including Bayesian learning [43], decision tree [26], boosting techniques [40], discriminant analysis [50], [19], dimension reduction [38], [50], ensemble learning [16], [39], etc. Moreover, some unsupervised learning techniques, like SOM [24] and EM algorithms [45], were also studied in the literature. Recently, Support Vector Machines (SVMs) [42] have been widely explored in machine learning, which enjoy superior performance in the real-world applications of pattern classification. A lot of research work has applied SVMs to relevance feedback in CBIR [41], [47], [18], [15]. Previous studies have shown that SVM is one of the most promising and successful approaches for attacking the relevance feedback problem.

1.3 Motivation of Our Work

Given the difficulty in learning the users' information needs from their feedback, multiple rounds of relevance feedback are usually required before satisfactory results are achieved. As a result, the relevance feedback phase can be extremely time-consuming. Moreover, the procedure of specifying the relevance of images in relevance feedback is usually viewed as a tedious and boring step by most users. Hence, it is required for a CBIR system with relevance feedback to achieve satisfactory results within as few feedback steps as possible, preferably in only one step. Despite previous efforts to accelerate relevance feedback using active learning techniques [41], traditional relevance feedback techniques are ineffective when the relevant samples are scarce in the initial retrieval results. From a long-term learning perspective, log data of accumulated users' relevance feedback could be used as an important resource to aid the relevance feedback task in CBIR. Although there have been a few studies carried out on the exploitation of users' log data in document retrieval [1], [9], little research effort has been dedicated to the relevance feedback problem in CBIR [17]. To our best knowledge, there has been no comprehensive work on integrating log of users' feedback into the learning process of relevance feedback in CBIR. Several recent studies related to our work are either too heuristic or lacking empirical evaluations from real-world users [14], [13], [49]. For example, the work in [13] suggested learning a semantic space by mining the relevance feedback log in CBIR. However, only the positive feedback was considered; the negative feedback examples, which can also be informative to users' information needs, were ignored.

In this paper, we present a novel framework for integrating the log data of users' relevance feedback with regular relevance feedback for image retrieval. In our framework, we compute the relevance information between query images and images in the database using both the log data and the low-level features of images and combine them to produce a more accurate estimation of relevance score. In order to make the learning algorithm more robust to erroneous log data in real-world applications, we propose a novel support vector machine (SVM) algorithm, named Soft Label SVM, to tackle the noisy data problem.

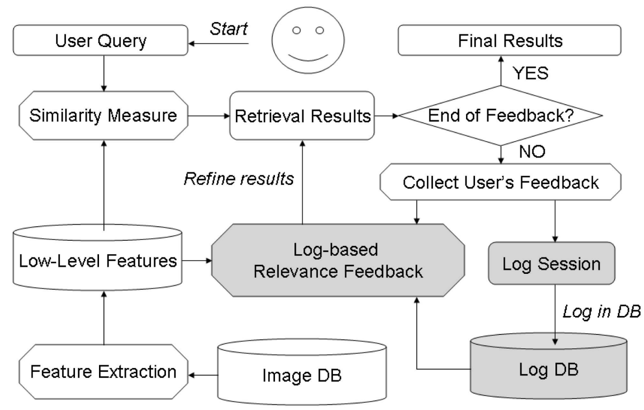


Fig. 1. The architecture of our proposed system.

The rest of this paper is organized as follows: Section 2 provides an overview of our framework for the log-based relevance feedback problem, followed by a formal definition and a unified solution for the problem. Section 3 gives a background review of SVMs from the regularization perspective and presents the Soft Label SVM that will be used to solve the log-based relevance feedback problem. Section 4 presents a log-based relevance feedback algorithm based on the Soft Label SVM technique. Section 5 discusses our experimental testbed and the methodology for performance evaluation of the log-based relevance feedback algorithm. Section 6 describes our empirical results for the log-based relevance feedback algorithm. Section 7 addresses the limitation of our scheme and the challenging problems for our algorithm, as well as the possible solutions in our future work. Section 8 concludes this work.

2 A Unified Log-Based Relevance Feedback Framework

2.1 Overview of Our Framework

We first give an overview of our proposed framework for log-based relevance feedback that systematically integrates the log data of users' relevance judgments with regular relevance feedback for image retrieval. Fig. 1 shows the architecture of the proposed system. First, a user launches a query in a CBIR system for searching desired images in databases. Then, the CBIR system computes the similarity between the user query and the image samples in database using the low-level image features. Images with high similarity measure are returned to the user. Next, the user judges the relevance of the initially returned results and submits his or her judgements to the CBIR system. A relevance feedback algorithm refines the initial retrieval results based on the user's relevance judgments, and returns an improved set of results to the user. Typically, a number of rounds of users' relevance feedback are needed to achieve satisfactory results.

Unlike traditional relevance feedback, we propose a unified framework that combines the feedback log with the regular relevance feedback. In Fig. 1, we see that the online relevance feedback from users is collected and stored in a log database. When feedback log data is unavailable, the log-based relevance feedback algorithm behaves exactly like a regular relevance feedback algorithm, which learns the correlation between low-level features and users'

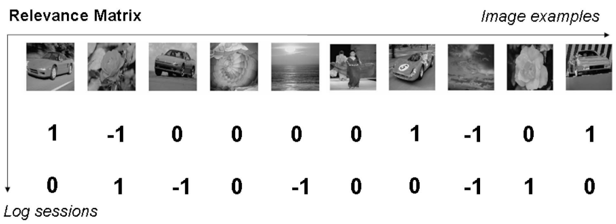


Fig. 2. The relevance matrix for representing the log information of user feedback. Each column of the matrix represents an image example in the image database and each row of the matrix corresponds to a log session in the log database.

information needs through the feedback image examples. When feedback log data is available, the algorithm will learn such a correlation using both the feedback log data and the online feedback from users. Thus, the log-based relevance feedback scheme is able to accomplish the retrieval goal in only a few iterations with the assistance from the log data of users' feedback.

2.2 Log-Based Relevance Feedback: Formulation and Definition

Before formally describing the problem of log-based relevance feedback, we need to systematically organize the log data of users' feedback. Assume a user labels N images in each round of regular relevance feedback, which is called a *log session* in this paper. Thus, each log session contains N evaluated images that are marked as either "relevant" or "irrelevant." For the convenience of representation, we construct a relevance matrix ($\mathbf{R}$) that includes the relevance judgements from all log sessions. Fig. 2 shows an example of such a matrix. In this figure, we see that each column of a relevance matrix represents an image example in the image database, and each row represents a log session from the log database. When an image is judged as "relevant" in a log session, the corresponding cell in matrix $\mathbf{R}$ is assigned to the value $+1$. Similarly, -1 is assigned when an image is judged as "irrelevant." For images that are not judged in a log session, the corresponding cells in $\mathbf{R}$ are assigned to zero values.

Based on the above formulation, we now define the log-based relevance feedback problem. Let us first introduce the following notation:

- $\mathbf{q}$: a user query.
- N_l: the number of labeled images for every log session.
- N_{img}: the number of image samples in the image database.
- N_{log}: the number of log sessions in the log database.

To retrieve the desired images, a user must first present a query $\mathbf{q}$, either by providing a query image or by drawing a sketch picture. Let $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_{N_{img}}\}$ denote the identity of images in the image database. Let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{N_{img}})$ denote the image database, where each $\mathbf{x}_i$ is a vector that contains the low-level features of the image $\mathbf{z}_i$. Let $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_{N_{log}})^T$ denote the log data in the log database, where each $\mathbf{r}_i$ contains relevance judgements in the ith log session. Let $\mathcal{L} = \{(\mathbf{z}_1, y_1), (\mathbf{z}_2, y_2), \ldots, (\mathbf{z}_{N_l}, y_{N_l})\}$ be the collection of labeled images acquired through the online feedback for a user. Then, the definition of a log-based relevance feedback problem can be given as follows:

Definition 1. *Log-Based Relevance Feedback. A log-based relevance feedback problem for image retrieval is to look for a relevance function $f_{\mathbf{q}}$ that maps each image sample $\mathbf{z}_i$ to a real value of relevance degree within 0 and 1,*

$$f_{\mathbf{q}} : \mathcal{Z} \longmapsto [0, 1],$$

based on the feature representation of images $\mathbf{X}$, the log data of users' feedback $\mathbf{R}$, and the labeled images $\mathcal{L}$ acquired from online feedback.

According to the above definition, both the low-level features of the image content, i.e., $\mathbf{X}$, and the log data of users' feedback, i.e., $\mathbf{R}$, should be included to determine the relevance function $f_{\mathbf{q}}$. Meanwhile, to reduce the number of iterations of online relevance feedback, a good learning algorithm should require only a small number of labeled image examples from the online relevance feedback, i.e., $|\mathcal{L}|$.

2.3 Solution to the Problem

Given that the relevance function depends on both $\mathbf{R}$ and $\mathbf{X}$, a simple strategy is to first learn a relevance function for each of these two types of information, and then combine them through a unified scheme. Let $f_{\mathbf{R}}(\mathbf{z}_i)$ denote a relevance function based on the log data of users' feedback and $f_{\mathbf{X}}(\mathbf{z}_i)$ denote a relevance function based on the low-level features of the image content. Both of them are normalized to $[0, 1]$, respectively. Then, the overall relevance function can be the combination of these two functions as follows:

$$f_{\mathbf{q}}(\mathbf{z}_i) = \frac{1}{2}(f_{\mathbf{R}}(\mathbf{z}_i) + f_{\mathbf{X}}(\mathbf{z}_i)). \tag{1}$$

In the following, we will describe how to acquire the relevance functions $f_{\mathbf{R}}(\mathbf{z}_i)$ and $f_{\mathbf{X}}(\mathbf{z}_i)$ separately.

Let us first consider the log data of users' feedback. When two images have similar content, we would expect different users to express similar relevance judgements for these two images. On the other hand, for two images with dramatically different content, there should be no correlation in their relevance judgments in log data. Hence, to estimate the similarity between two images $\mathbf{z}_i$ and $\mathbf{z}_j$, we suggest a modified correlation function to measure their relevance judgments in the log data, i.e.,

$$c_{i,j} = \sum_k \delta_{k,i,j} \cdot r_{k,i} \cdot r_{k,j}, \tag{2}$$

where $\delta_{k,i,j}$ is defined as follows:

$$\delta_{k,i,j} = \begin{cases} 1 & \text{if} \quad r_{k,i} + r_{k,j} \geq 0, \\ 0 & \text{if} \quad r_{k,i} + r_{k,j} < 0. \end{cases} \tag{3}$$

Note that $\delta_{k,i,j}$ is engaged to remove $(-1, -1)$ pairs among $(r_{k,i}, r_{k,j})$ in the computation of similarity. This is because it is difficult to judge the similarity of two images when they both are marked as "irrelevant" to users' information needs. Evidently, image $\mathbf{z}_i$ and image $\mathbf{z}_j$ are relevant when c_{ij} is positive, irrelevant when c_{ij} is negative. When c_{ij} is around zero, it is usually hard to judge if one image is relevant to the other.

Based on the above similarity function, we can develop the relevance function based on the log data. Let $\mathcal{L}^+$ denote the set of positive (or relevant) images in $\mathcal{L}$, and $\mathcal{L}^-$ denote the set of negative (or irrelevant) samples. For an image in the database, we compute its overall similarities to both positive and negative images, and the difference between these two similarities will indicate the relevance of the image to the user's query. More specifically, the overall relevance function can be formulated as follows:

$$f_{\mathbf{R}}(\mathbf{z}_i) = \max_{k \in \mathcal{L}^+} \left\{ \frac{c_{k,i}}{\max_j c_{k,j}} \right\} - \max_{k \in \mathcal{L}^-} \left\{ \frac{c_{k,i}}{\max_j c_{k,j}} \right\}. \tag{4}$$

Despite its simple form, our empirical studies have shown that the above relevance function is effective in practice [17].

Remark. So far, we assume the above relevance function is calculated on the fixed log data. Toward a long-term learning purpose, it is important to develop an incremental method to deal with the new added log session. For the method proposed above, it is natural to provide an incremental solution. For example, we can create a correlation matrix $C_M = [c_{i,j}]_{N_{img} \times N_{img}}$ which marks down the correlation values between images based on the history log data. When a new log session is added to the log database, we can update the element in the correlation matrix as follows:

$$c_{i,j} = c_{i,j} + \delta_{k',i,j} \cdot r_{k',i} \cdot r_{k',j}, \tag{5}$$

where $\mathbf{r}_{k'}$ is the new log session, $\delta_{k',i,j}$ is defined in (3). Note that only the element $c_{i,j}$ satisfying $r_{k',i} \neq 0$ and $r_{k',j} \neq 0$ will be updated.

After obtaining the relevance function on the log data, we can use it in learning the relevance function on the low-level image features. Learning the relevance function on the image features is a standard relevance feedback problem in content-based image retrieval. Dozens of suitable algorithms have been proposed in the literature [19]. Among them, support vector machine (SVM) is one of the most effective techniques in practice. As a state-of-the-art classification technique, SVM enjoys excellent generalization capability which has shown superior performance in many applications. Although it is able to function with small numbers of training samples, the performance of SVM will usually deteriorate significantly when the number of training samples is too small. This is a general issue with any discriminative classifier as pointed out in [28]. Given that the number of labeled samples in $\mathcal{L}$ is small, applying SVM directly to $\mathcal{L}$ may not achieve the desirable performance. One possible solution is to boost the performance using unlabeled samples by the Transductive SVM [23]. However, difficulties such as high training cost [23] and unstable performance [48] prevent its application to the relevance feedback problem.

Hence, we propose enriching training samples by employing the relevance function based on the log data in (4). One simple approach is to calculate the relevance scores of image samples to the query target using (4) and augment training examples with the image samples that have large relevance scores. Although this approach can be straightforwardly handled by the standard SVM algorithm, it may suffer from performance degradation, providing that image samples with high relevance scores may not be relevant to the targeted query. To deal with this noisy data problem, we propose a novel learning algorithm, named Soft Label Support Vector Machine. Unlike the standard SVMs in which all the training examples are labeled as either "+1" or "-1," our algorithm does not require absolute confidence about the labels of the selected training samples. In fact, the relevance scores of images reflect the uncertainties in determining their labels. Thus, instead of using hard binary labels, we introduce the "soft label" for the training samples that use the relevance scores computed from (4). By combining the soft-labeled samples with the labeled samples acquired from the online user feedback, we can train a Soft Label SVM classifier. The final relevance function on the low-level image features will be constructed based on the decision function of the trained classifier. In the following section, we first introduce the background of SVM and then formulate the Soft Label SVM technique in detail.

3 Soft Label Support Vector Machines

3.1 Overview of Regularization Framework and Support Vector Machines

Support Vector Machines (SVMs) enjoy solid theoretical foundations and have demonstrated outstanding performance in many empirical applications [5]. In theory, SVM can be interpreted from the solid regularization theory framework which has been used in many machine learning problems [10]. In order to provide a rigorous justification of Soft Label Support Vector Machine, we here provide a brief overview of regularization framework and Support Vector Machines.

In a general setting of learning from examples, we are given a training set of l independent and identically distributed observations

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l),$$

where $\mathbf{x}_i$ are vectors produced by a generator and y_i are the associated responses by a supervisor. A learning machine estimates a set of approximated functions f to approach the supervisor's responses. It is an ill-posed problem to approximate a function from sparse data, which is solved by the regularization theory in a typical way [10]. The classical regularization theory formulates the learning problem as a variational problem of finding the function f which tends to minimize the following functional:

$$f = \arg\min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} (y_i - f(\mathbf{x}_i))^2 + \lambda \|f\|_K^2, \tag{6}$$

where $\|f\|_K^2$ is a norm in a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ defined over the positive definite function K and λ is the regularization parameter.

The classical regularization theory has been justified by the significant work of Vapnik's theory [42]. Based on the framework of Vapnik's theory, a more general regularization framework is suggested to find the function f via the functionals

$$f = \arg\min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} V(\mathbf{x}_i, y_i, f) + \lambda \|f\|_K^2, \tag{7}$$

where $V(\cdot,\cdot,\cdot)$ is a loss function, and the penalty norm $\lambda\|f\|_K^2$ imposes smoothness conditions on the solution space. Based on the above regularization framework, many well-known algorithms, such as regularized networks, support vector machine regression, and support vector machine classification can be interpreted in terms of different loss functions. Here we give three different choices of loss functions which correspond to three state-of-the-art algorithms:

- Regularized Least Squares Networks (RLS):
$$V(\mathbf{x}_i, y_i, f) = (y_i - f(\mathbf{x}_i))^2, \tag{8}$$
- Support Vector Machine Regression (SVMR):
$$V(\mathbf{x}_i, y_i, f) = (y_i - f(\mathbf{x}_i))_\epsilon, \tag{9}$$
- Support Vector Machine Classification (SVMC):
$$V(\mathbf{x}_i, y_i, f) = (1 - y_i f(\mathbf{x}_i))_+, \tag{10}$$

where $(\cdot)_\epsilon$ is the Vapniks epsilon-insensitive norm [42], $(\cdot)_+$ is the hinge loss in which $(a)_+ = a$ if a is positive and zero otherwise, and y_i is a real number for both RLS and SVMR, and takes $+1$ or -1 for SVMC. To avoid confusion, we limit our further discussion on classification and retrieval problems.

The loss function in (10) is also named as the soft margin loss function for SVM classification. But, SVM practitioners may be familiar with another alternative formulation involving the C parameter as follows:

OPT 1—Standard SVM (Soft Margin Nonseparable)

$$\begin{aligned} \min_{\mathbf{w},\xi,b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i) - b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, 2, \cdots, l, \end{aligned} \tag{11}$$

where C is a regularization parameter, which is equivalent to $\frac{1}{2\lambda l}$ where λ is the parameter in the above regularization framework, and $\Phi(\cdot)$ is a kernel mapping function, labels y_i are either $+1$ or -1 for a regular binary classification problem.

3.2 Soft Label Support Vector Machines

According to the regularization framework in (7), it is critical to define an appropriate loss function that fits in with the nature of the application. In standard SVMs for classification applications, the given training samples are normally assumed noise-free. When this assumption is not satisfied, the original loss function may not be the best choice. This motivates us to study the Soft Label Support Vector Machines for the cases of noisy labels. To facilitate the following discussion, we denote the regular support vector machine for noise-free cases as "Hard Label Support Vector Machine" (SVM), and the noise-appearing cases as "Soft Label Support Vector Machine" (SLSVM).

Suppose we are given the training data as follows:

$$(\mathbf{x}_1, s_1), (\mathbf{x}_2, s_2), \ldots, (\mathbf{x}_m, s_m),$$

where the label s_i is a real number and $0 < |s_i| < 1$.[1] In the above setting, the sign of each label s_i, i.e., $sgn(s_i)$, indicates the binary class label of the corresponding sample. The magnitude of label s_i, i.e., $|s_i|$, represents the confidence of the assigned label. We call these labels "soft labels" to distinguish them from the binary labels. Our goal is to learn a reliable SVM classification model from the data points that are "softly" labeled.

A straightforward approach is to convert a Soft Label learning problem into the one with hard labels. However, this will discard the confidence information related to the soft labels, which may significantly degrade the performance of the classifier. In order to develop a more robust scheme for exploiting the information of soft labels, we propose to modify the loss function of SVMs in (10). Our first formal definition of the Soft Label loss function is given as follows:

$$V(\mathbf{x}_i, s_i, y_i, f) = |s_i| \cdot (1 - y_i f(\mathbf{x}_i))_+. \tag{12}$$

Different from (10), the loss term is weighted by $|s_i|$, i.e., the confidence of the assigned label. The larger the confidence $|s_i|$ is, the more important the loss term of the sample will be. We further expand the loss function defined in (12) by including the hard-labeled data, i.e.,

$$V(\mathbf{x}_i, s_i, y_i, f) = \begin{cases} C_H \cdot (1 - y_i f(\mathbf{x}_i))_+ & \text{if } |s_i| = 1, \\ C_S \cdot |s_i| \cdot (1 - y_i f(\mathbf{x}_i))_+ & \text{if } 0 < |s_i| < 1. \end{cases} \tag{13}$$

In the above definition, we assume the hard-labeled data points correspond to the case when $|s_i| = 1$. Two weight parameters C_H and C_S are introduced to balance the importance between hard-labeled data and soft-labeled data. Usually, we set $C_H > C_S > 0$. This is based on the intuition that the cost of misclassifying a hard-labeled example should be significantly higher than the cost of misclassifying a softly labeled example. By carefully choosing the value of C_S and C_H, our SVM algorithm is able to, on the one hand, fully take advantage of the soft-labeled examples to narrow down the best location for the decision boundary, and on the other hand, avoid being misled by the potentially erroneous labels in the soft-labeled data.

Now, assume $f(\mathbf{x}) = \mathbf{w}\cdot\Phi(\mathbf{x}) - b$. By substituting the definition of loss function in (13) into the general framework in (7), we have

$$\begin{aligned} \min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 &+ C_H\sum_{i=1}^{l}(1 - y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i) - b))_+ \\ &+ C_S\sum_{i=l+1}^{l+m}|s_i|(1 - y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i) - b))_+. \end{aligned} \tag{14}$$

To simplify the above problem, we introduce a slack variable $\xi_i = (1 - y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i) - b))_+$ for every labeled example (including both hard-labeled instances and soft-labeled instances), which leads to the following optimization problem:

1. If a training sample is given with $s_i = 0$, it will be treated as an unlabeled data instance which is excluded from our learning machine.

OPT 2—Soft Label Support Vector Machine

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C_H \sum_{i=1}^{l} \xi_i + C_S \sum_{i=l+1}^{l+m} |s_i|\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) - b) \geq 1 - \xi_i, \\ & \xi_i \geq 0 \,, i = 1, \ldots, l+m, \end{aligned} \tag{15}$$

where l and m are, respectively, the number of hard-labeled training data and the number of soft-label ones (with $|s_i| < 1$), C_H and C_S are weight parameters for hard-labeled and soft-labeled training data, respectively. For softly labeled examples, $y_i = sgn(s_i)$. Note that when all $|s_i| = 0$, the above optimization problem is reduced to a standard SVM in **OPT 1**.

The solution to the above optimization problems can be found by introducing the Lagrange functional technique, similar to the method of solving standard SVMs [42]. Here, we simply state the final result:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{l+m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l+m} \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \\ \text{subject to} \quad & \sum_{i=1}^{l+m} \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C_H, i = 1, 2, \ldots, l, \\ & 0 \leq \alpha_i \leq |s_i| C_S, i = l+1, l+2, \ldots, l+m. \end{aligned} \tag{16}$$

More details are referred to the appendix. Notice that the upper bounds of the weights α_i for softly labeled examples are proportional to the confidence of their class labels. As a result, the misclassification cost is directly proportional to the confidence of labeling examples. Apparently, this is consistent with our common intuition. Similar to standard SVMs, the optimization problem in (16) is a typical quadratic programming problem that can be solved effectively by available techniques [29].

4 Log-Based Relevance Feedback Using Soft Label SVM

In Section 2, we provide a unified framework for developing a log-based relevance feedback algorithm in general. The key idea is to first identify a relevance function based on the log data of users' feedback, i.e., $f_R(\mathbf{x})$. Then, the log-based relevance function is used to aid the learning task of the relevance function based on the low-level image features, i.e., $f_X(\mathbf{x})$. Finally, these two relevance functions are combined together to rank all the images. Given the erroneous log data, applying traditional techniques to the log-based relevance feedback may be problematic on account of the noise in the data.

To develop an effective log-based relevance feedback algorithm, a modified SVM technique, i.e., the Soft Label SVM, was proposed in the preceding section to attack the noise problem. In contrast to standard SVMs, the Soft Label SVMs incorporates the label confidence into the learning task. In this section, we develop a practical algorithm for log-based relevance feedback using Soft Label SVM, which we refer to as LRF-SLSVM. It can be summarized in four steps as follows:

1. *Calculate relevance scores $f_R(\mathbf{z})$ for all image samples.* The relevance scores are computed using (4) to evaluate the initial relevances of images in the database based on the log data. Despite its simple form, (4) is empirically effective.
2. *Choose training samples with Soft Labels based on their relevance scores*. Image samples with large relevance scores obtained in Step 1 will be chosen as pseudo-training samples and their relevance scores are normalized to serve as the "soft label" for Soft Label SVM.
3. *Train a Soft Label SVM classifier on the selected training samples with Soft Labels, i.e., $f_{SLSVM}(\mathbf{z})$.* Given the labeled samples acquired from online feedback and the softly labeled examples acquired in Step 2, a Soft Label SVM classifier is trained according to **OPT 2** in (15).
4. *Rank images based on the combination of the two relevance functions $f_R(\mathbf{z})$ and $f_{SLSVM}(\mathbf{z})$.* The two relevance functions $f_R(\mathbf{z})$ and $f_{SLSVM}(\mathbf{z})$ will first be normalized and then combined together to form the overall relevance function, i.e., $f_q(\mathbf{z}) = f_R(\mathbf{z}) + f_{SLSVM}(\mathbf{z})$.

Fig. 3 provides the pseudocode of the algorithm of log-based relevance feedback by Soft Label SVM, in which the relevance function $f_R(\mathbf{z})$ is represented by $(R_p(\mathbf{z}) - R_n(\mathbf{z}))$. Implementation details of the proposed algorithm will be discussed in the following experimental section.

5 Experimental Methodology

5.1 Overview of Experimental Testbeds

The experimental testbeds and settings are critical to evaluating the performance of log-based relevance feedback algorithms. So far, there is not a benchmark data set available for the log-based relevance feedback problem. Thus, we must design a set of objective and practical experimental testbeds which not only accurately evaluate our algorithms but also adequately facilitate real-word applications.

As we have known, empirical evaluation of a CBIR system by humans may be somewhat subjective. Hence, it is necessary to develop an automatic mechanism to evaluate the retrieval performance of CBIR. However, several previous studies on log-based relevance feedback simply generate user data through simulations, which may not reflect the true challenges of real-world applications. To address this problem, in our experiment, a testbed is carefully built to allow for the objective evaluation of content-based image retrieval, while maintaining close analogy to real-world applications. In particular, our testbeds include three components: image data sets, low-level image representation, and the collection of users' log data.

5.2 Image Data Sets

To perform empirical evaluation of our proposed algorithm, we choose the real-world images from the COREL image CDs. There are two sets of data used in our

Algorithm: **LRF-SLSVM**

Input:

$\mathbf{q}$ /* a query sample by a user */

$\mathcal{L}$ /* set of labeled training samples */

Variables:

$\mathcal{S}$ /* set of "Soft Label" training samples */

C_H, C_S /* regularization parameters in $\mathbf{OPT\ 2}$ */

c /* correlations or relationships between images */

R_p, R_n, f_R /* log-based relevance degrees to the query */

Δ /* selection threshold for "Soft Label" samples */

f_{SLSVM} /* a Soft Label SVM classifier */

f_q /* the overall relevance function */

Output:

$\mathcal{R}_{top}$ /* set of most relevant samples */

BEGIN

/* *Step. (1) compute log-based relevance functions* */

for each positive $\mathbf{z} \in \mathcal{L}\{$

for each $\mathbf{z}_i \in \mathcal{Z}\{$

c(i) $\leftarrow$ CompRelationship($\mathbf{z}$, $\mathbf{z}_i$); } /* by Equation (2) */

c $\leftarrow$ Normalize(c); /* normalize to [0, 1]*/

R_p(i) $\leftarrow$ max(R_p(i), c(i)); } /* Init: $R_p(i) \leftarrow -\infty$ */

for each negative $\mathbf{z} \in \mathcal{L}\{$

for each $\mathbf{z}_i \in \mathcal{Z}\{$

c(i) $\leftarrow$ CompRelationship($\mathbf{z}$, $\mathbf{z}_i$); } /* by Equation (2) */

Normalize(c); /* normalize to [0, 1]*/

$R_n(i) \leftarrow$ max(R_n(i), c(i)); } /* Init: $R_n(i) \leftarrow -\infty$ */

/* *Step. (2) select "Soft Label" training samples* */

for each $\mathbf{z}_i \in \mathcal{Z}\{$

if $R_p(i) - R_n(i) \geq \Delta$, **then** $\mathcal{S} \leftarrow \mathcal{S} \bigcup \{\mathbf{z}_i\}$; }

/* *Step. (3) train a Soft Label SVM classifier* */

$f_{\mathrm{SLSVM}} \leftarrow$ Train_Soft_Label_SVM($\mathcal{L}$, $\mathcal{S}$, C_H, C_S)

$f_{\mathrm{SLSVM}} \leftarrow$ Normalize(f_{SLSVM});

/* *Step. (4) rank images based on* f_{SLSVM} *and* $(R_p - R_n)$ */

$f_R \leftarrow$ Normalize($R_p - R_n$);

$f_q \leftarrow f_{\mathrm{SLSVM}}$ + f_R;

$\mathcal{R}_{top} \leftarrow$ Sort_In_Decend_Order(f_q);

return $\mathcal{R}_{top}$;

END

Fig. 3. The algorithm of log-based relevance feedback by Soft Label SVM.

experiments: 20-Category (20-Cat) that contains images from 20 different categories and 50-Category (50-Cat) that includes images from 50 categories. Each category in the data sets consists of exactly 100 images that are randomly selected from relevant examples in the COREL image CDs. Every category represents a different semantic topic, such as *antique*, *antelope*, *aviation*, *balloon*, *botany*, *butterfly*, *car*, *cat*, *dog*, *firework*, *horse*, and *lizard*, etc.

The motivation for selecting images in semantic categories is twofold. First, it allows us to evaluate whether the proposed approach is able to retrieve the images that are not only visually relevant but also semantically similar. Second, it allows us to evaluate the retrieval performance automatically, which will significantly reduce the subjective errors relative to manual evaluations.

5.3 Low-Level Image Representation

Image representation is an important step in the evaluation of relevance feedback algorithms in CBIR. Three different sets of features are chosen in our experiments to represent the images: color, edge, and texture.

Color features are widely adopted in CBIR for their simplicity. The color feature extracted in our experiments is the color moment. It is close to natural human perception, whose effectiveness in CBIR has been shown in many previous research studies. Three different color moments are used: color mean, color variance, and color skewness in each color channel (H, S, and V), respectively. Thus, a nine-dimensional color moment is adopted as the color feature.

Edge features can be very effective in CBIR when the contour lines of images are evident. The edge feature used in our experiments is the edge direction histogram [21]. To acquire the edge direction histogram, an image is first translated to a gray image, and a Canny edge detector is applied to obtain its edge image. Based on the edge images, the edge direction histogram can then be computed. Each edge direction histogram is quantized into 18 bins of 20 degrees each. Hence, an 18-dimensional edge direction histogram is employed to represent the edge feature.

Texture features are proven to be an important cue for image retrieval. In our experiments, we employ the wavelet-based texture technique [27], [37]. A color image is first transformed to a gray image. Then, the Discrete Wavelet Transformation (DWT) is performed on the gray image using a Daubechies-4 wavelet filter [37]. Each wavelet decomposition on a gray 2D-image results in four subimages with a $0.5 * 0.5$ scaled-down image of the input image and the wavelets in three orientations: horizontal, vertical, and diagonal. The scaled-down image is then fed into the DWT to produce the next four subimages. In total, we perform a three-level decomposition and obtain 10 subimages in different scales and orientations. One of the 10 subimages is a subsampled average image of the original image and, thus, is discarded. For the other nine subimages, we compute the entropy of each subimage separately. Hence, a wavelet-based texture feature of nine dimensions in total is computed to describe the texture information of each image.

In sum, a 36-dimensional feature vector is used to represent an image, including nine-dimensional color histogram, 18-dimensional edge direction histogram, and nine-dimensional wavelet-based texture.

5.4 Log Data Collection of User Feedback

Collecting the log data of users' feedback is an important step for a log-based relevance feedback scheme. In our experiment, we have developed a CBIR system with a relevance feedback mechanism to collect the relevance feedback from real-world users. Fig. 4 shows the Graphical

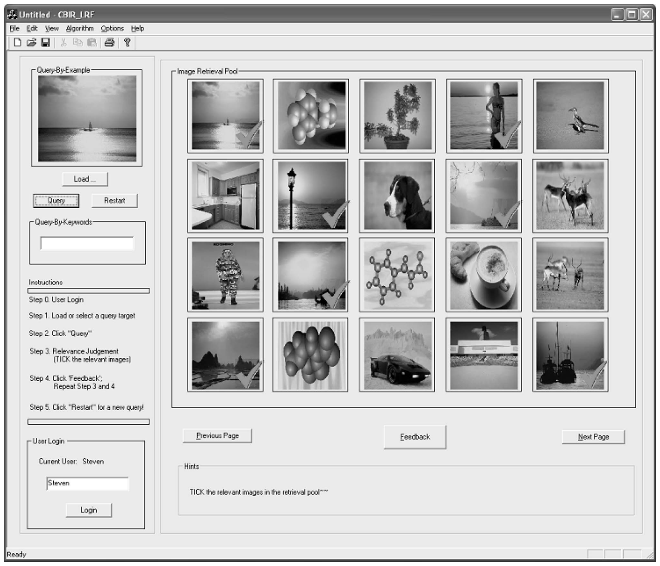


Fig. 4. The GUI of our CBIR system with relevance feedback. A user can simply TICK the relevant images from the retrieval pool to provide his/her feedback. The ticked images are logged as positive samples; others are regarded as negative samples.

User Interface (GUI) of our CBIR system for collecting feedback data. Through the GUI, a user can provide his or her relevance judgements by simply ticking relevant images from the retrieval pool. We describe the details on the collection of the feedback log data and the definition on the format of the log data as follows:

For a retrieval task in CBIR, a user begins a query session by presenting a query example. In our experiment, a user will first randomly select a query image from the image database as the query goal. Then, the user submits the query example to the CBIR system and obtains a set of initial retrieval results from the CBIR system after a query-by-example execution. Based on the retrieval results, the user can tick the relevant images in the retrieval pool. After the relevant samples are ticked in a relevance feedback session, the user can submit his or her judgement results to the CBIR system, in which the feedback results will be stored in the log database. To quantitatively analyze the retrieval performance, we define a log session as the basic unit of the log data. Each log session corresponds to a regular relevance feedback session, in which 20 images are judged by the user. Thus, each log session contains 20 labeled images that are marked as either "relevant (positive)" or "irrelevant (negative)."

One important issue with the log data is its noise problem, which is caused by the subjective judgments from the human subjects involved in our study. Given the fact that different users are likely to have different opinions on judging the same image, the noise problem in log-based relevance feedback is inevitable in real-world applications. In order to evaluate the robustness of our algorithm, we collect log data with different amount of noise. The noise of log data is measured by its percentage of incorrect relevance judgments P_{noise}, i.e.,

$$P_{noise} = \frac{\text{Total number of wrong judgements}}{N_l \times N_{log}} \times 100\%, \quad (17)$$

TABLE 1
The Log Data Collected from Users on Both Data Sets

Datasets	Small Noise Log Data		Large Noise Log Data	
	# Log Sessions	P_{noise}	# Log Sessions	P_{noise}
20-Category	100	7.8%	100	16.2%
50-Category	150	7.7%	150	17.1%

where N_l and N_{log} stand for the number of labeled examples acquired for each log session and the number of log sessions, respectively.

In our experiment, 10 users help us collect the log data using our CBIR system. Two sets of log data with different amount of noise are collected on both data sets in the experiment: log data with low noise that contains fewer than 10 percent of incorrect relevance judgments and log data with high noise that contains more than 15 percent of incorrect relevance judgments. Table 1 shows the two sets of collected log data for both data sets with different amounts of noise from real-world users. In total, 100 log sessions are collected for the 20-Category data set and 150 log sessions for the 50-Category data set. Based on these log data with different configurations, we are able to evaluate the effectiveness, the robustness, and the scalability of our proposed algorithm.

6 EXPERIMENTAL RESULTS

6.1 Overview of Performance Evaluation

The experiments are designed to answer the following questions:

1. *Are log-based relevance feedback schemes more effective than traditional relevance feedback methods?* To this end, we compare the performance of log-based relevance feedback algorithms with that of traditional relevance feedback algorithms. Two relevance feedback algorithms are used as our baseline, namely, the query expansion approach and the classification approach based on support vector machines.
2. *Is the proposed algorithm for log-based relevance feedback more effective than other alternatives*? To address this question, we will compare the Soft Label SVM based approach for log-based relevance feedback to other approaches that also utilize the log data to improve the performance of image retrieval. The two methods included in this study are the query expansion-based approach and the SVM-based approach.
3. *Is the Soft Label SVM-based approach more resilient to noisy log data than the standard SVM based approach?* The noise problem is inevitable in log data. To examine the robustness of the proposed algorithm, we evaluate the performance of the Soft Label SVM-based approach against log data with different levels of noise and compare it with the log-based relevance feedback approach that engages the standard SVM. Since the choice of two weight parameters C_S and C_H can have significant impact on the final retrieval

results, we also conduct experiments with different C_S and C_H to see how they affect the robustness of the proposed Soft Label SVM.

6.2 The Compared Schemes

In our compared schemes, a simple Euclidean distance measure approach (RF-EU) serves as the baseline method. Two traditional relevance feedback schemes are engaged in our comparisons, i.e., relevance feedback by query expansion (RF-QEX) [30] and relevance feedback by support vector machine (RF-SVM) [41], [47]. In addition to the Soft Label SVM-based approach, we also develop two methods for log-based relevance feedback based on our suggested framework by using the traditional query expansion technique (LRF-QEX) and standard SVMs. The details of the compared schemes are given as follows:

6.2.1 Euclidean

This is recorded as a reference of performance comparison. In our approach, Euclidean distances between query images and images in the database are first measured, and images with small distances are then returned to the users. Despite the fact that there have been many other more sophisticated distance measures investigated in CBIR [2], the Euclidean distance scheme is employed in our experiment because of its simplicity and its robustness.

6.2.2 RF-QEX

Query expansion for relevance feedback originates from traditional information retrieval [46], [34]. A lot of different approaches have been proposed to formulate relevance feedback algorithms based on the idea of query expansion [31], [19]. Query expansion can be viewed as a multiple-instance sampling technique [20], in which the returned samples in the next round are selected from the neighborhood of the positive samples of the previous feedback round. Many previous studies have shown that query expansion is effective in relevance feedback for image retrieval [31]. In our experiment, we implement the similar relevance feedback approach in [31] for image retrieval. Specifically, given N_l samples labeled by a user in a relevance feedback round, the images with the smallest Euclidean distances to the N_l positive samples are retrieved to the results. Meanwhile, the negative labeled samples are excluded from the retrieval list if they fall in the selected nearest neighbor of any positive samples.

6.2.3 RF-SVM

Relevance feedback by support vector machine is one of the most popular and promising schemes used in image retrieval [15], [16], [41], [47]. In our experiment, we implement the SVM-based relevance feedback scheme using the Gaussian kernel.

6.2.4 LRF-QEX

Query expansion has been shown to be effective in exploiting user query log data in traditional document information retrieval [8]. In our experiment, we extend it to log-based relevance feedback for image retrieval. More specifically, log-based relevance feedback with query expansion can be described as follows: We first compute the relevance score $f_{\mathbf{R},L}(\mathbf{z}_i)$ for each image $\mathbf{z}_i$ using (4). Then, for each image in the database, and for every image $\mathbf{z}_j^+$ that is positively labeled by the user, we measure their Euclidean distance $f_{\mathrm{EU}}(\mathbf{z}_i, \mathbf{z}_j^+)$ based on the low-level image features. The final relevance score $f_{\mathbf{q}}(\mathbf{z}_i)$ for each image $\mathbf{z}_i$ is determined by the combination of $f_{\mathrm{EU}}(\mathbf{z}_i, \mathbf{z}_j^+)$ and $f_{\mathbf{R},L}(\mathbf{z}_i)$, i.e., $f_{\mathbf{q}}(\mathbf{z}_i) = f_{\mathbf{R},L}(\mathbf{z}_i) - \min_j f_{\mathrm{EU}}(\mathbf{z}_i, \mathbf{z}_j^+)$. Images with the largest relevance scores will be returned to the users. As with the query expansion approach for standard relevance feedback, images that are already labeled as negative will be excluded from the retrieval list.

6.2.5 LRF-SLSVM

The algorithm of the log-based relevance feedback by Soft Label SVM is given in Fig. 3. To train the Soft Label SVM classifier, similar to standard SVMs, we apply the sequential minimum optimization (SMO) approach [6].

6.2.6 LRF-SVM

To examine the effectiveness and robustness of the Soft Label SVM, we also implement a method for log-based relevance feedback using standard SVMs, which is similar to the algorithm in Fig. 3.

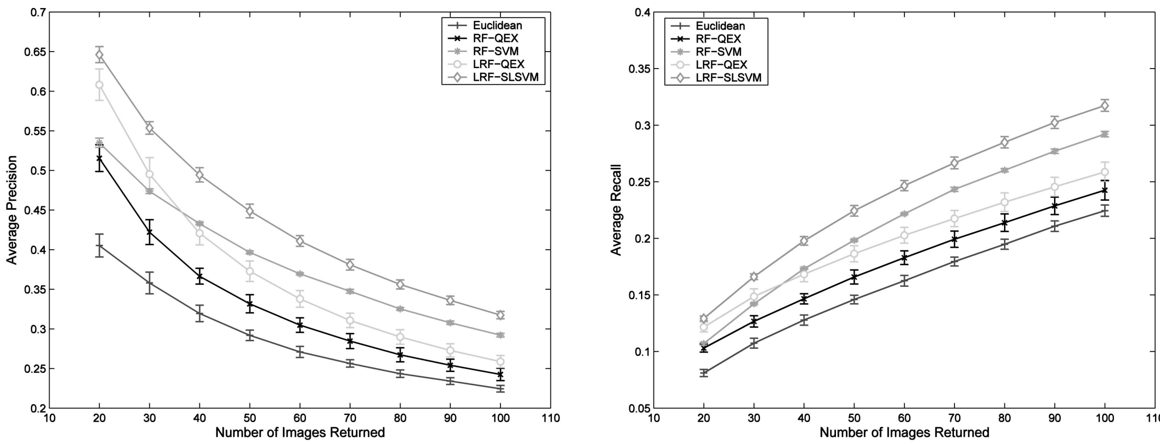
6.3 Experimental Implementation

The implementation of SVMs in our experiments is based on the public *LIBSVM* library available at [6]. To implement the Soft Label SVM algorithm, we modify the library based on the optimization in (16). It is a well-known fact that kernels and their parameters play an important role in the performance of SVMs. In our experiment, the Radial Basis Function (RBF) kernel is used in both the Soft Label SVM and standard SVMs, which is given as $K(\mathbf{x}, \mathbf{x}') = exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2)$, where γ is a positive constant. The reason for choosing the RBF kernel is that it has been shown to be very effective in multimedia retrieval problems in many previous studies [41], [15]. Besides the kernel selection, the choice of regularization parameters in the standard SVM and the Soft Label SVM is also critical to the retrieval performance. In our experimental implementation, the parameter C in the standard SVM and the two parameters C_H and C_S are chosen empirically using a separate validation data set.

For a retrieval task, it is important to define a suitable metric for performance evaluation. Two metrics are employed in our experiments as follows:

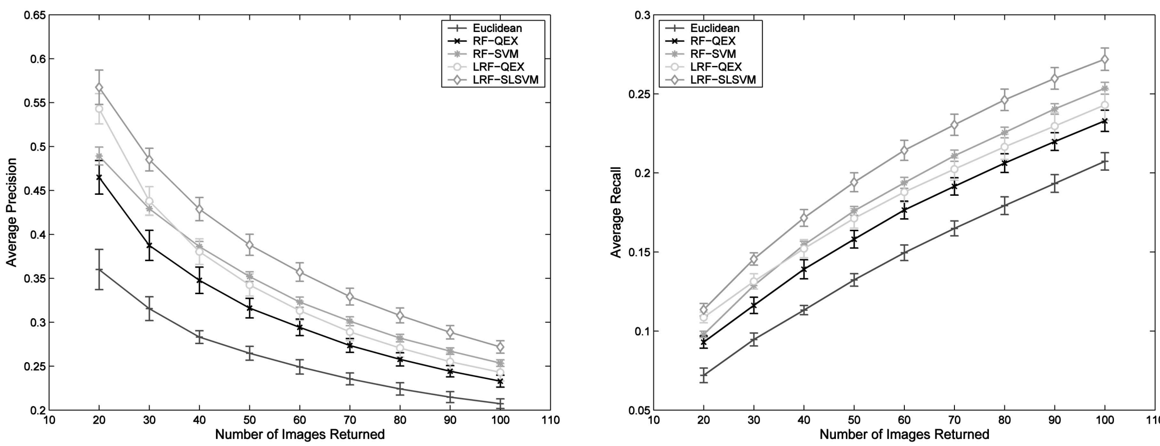
1. **Average Precision**, which is defined as the percentage of relevant images among all the images that have been retrieved, and
2. **Average Recall**, which is defined as the percentage of relevant images of retrieved images among all relevant images in the data set.

In our experiment, all compared schemes are evaluated on 200 queries randomly selected from the data set. The reported results of *Average Precision* and *Average Recall* are obtained by taking an average over the 200 queries. For each query, the number of labeled samples acquired from the online user feedback is 10, and the top 100 samples are returned to be evaluated for all compared schemes. To observe the overall performance, *Mean Average Precision* (MAP) is measured on top ranked images, ranging from the



(a) (b)

Fig. 5. Performance evaluation on the 20-Category data set with small noise log data. (a) Average precision. (b) Average recall.



(a) (b)

Fig. 6. Performance evaluation on the 50-Category data set with small noise log data. (a) Average precision. (b) Average recall.

top 20 images to the top 100 images. Finally, all the compared schemes are evaluated on both the 20-Category and the 50-category data sets.

The experimental platform is on Windows and all algorithms are implemented in MS Visual C++ for the purpose of efficiency. The hardware environment of all experiments is a PC machine with a 2.0G Pentium-4 CPU and 512MB memory.

6.4 Effectiveness of Our Log-Based Relevance Feedback Scheme

In order to verify the effectiveness of our log-based relevance feedback scheme, we evaluate two log-based relevance feedback algorithms and two traditional relevance feedback algorithms. The algorithms for traditional relevance feedback are the query expansion approach (RF-QEX) and the SVM approach (RF-SVM). The two algorithms for log-based relevance feedback include log-based relevance feedback by query expansion (LRF-QEX) and log-based relevance feedback by Soft Label SVM (LRF-SLSVM). These algorithms are evaluated on the log data with low noise, i.e., 7.8 percent noise for the 20-Category data set and 7.7 percent noise for the 50-Category data set.

Fig. 5 and Fig. 6 show the experimental results of the compared algorithms using this log data. The horizontal axis is the number of top ranked images used in evaluation, and the vertical axis is the *Average Precision* and *Average Recall* measured on the top ranked images. As these figures show, it is evident that the two log-based relevance feedback algorithms (LRF-QEX and LRF-SLSVM) substantially outperform the two algorithms with traditional relevance feedback (RF-QEX and RF-SVM). For example, on the 20-Category data set, the average precision of the LRF-QEX algorithm achieves an 18.0 percent improvement over the regular RF-QEX algorithm on the top 20 images. By contrast, the absolute improvement of the LRF-SLSVM algorithm over the regular RF-SVM algorithm is 20.8 percent on the top 20 images. With reference to the MAP on average, the LRF-QEX algorithm has an 11.7 percent improvement over the RF-QEX algorithm, and the LRF-SVM algorithm has a 12.6 percent improvement over the RF-SVM algorithm.

The results on the 50-Category data set are similar, but the improvement is slightly smaller than the 20-Category one. This is because the content of the 50-Category is more diverse than the 20-Category one, since the former contains more

TABLE 2
Performance Comparisons (Average Precision) on Different Amounts of Log Data on the 20-Category Data Set

Algorithm	TOP-20	TOP-40	TOP-60	TOP-80	TOP-100	MAP
RF-QEX (Baseline)	0.516±0.017	0.367±0.010	0.305±0.009	0.267±0.009	0.243±0.008	0.332±0.011
RF-SVM	0.535±0.006 (+3.8%)	0.433±0.002 (+18.1%)	0.370±0.001 (+21.2%)	0.325±0.002 (+21.7%)	0.292±0.002 (+20.4%)	0.387±0.002 (+17.8%)
LRF-QEX (#LS=50)	0.569±0.024 (+10.3%)	0.395±0.015 (+7.8%)	0.321±0.013 (+5.2%)	0.279±0.011 (+4.3%)	0.251±0.010 (+3.6%)	0.355±0.015 (+6.2%)
LRF-QEX (#LS=100)	0.608±0.020 (+18.0%)	0.421±0.015 (+14.8%)	0.338±0.010 (+10.9%)	0.290±0.009 (+8.5%)	0.259±0.008 (+6.7%)	0.374±0.013 (+11.7%)
LRF-SLSVM (#LS=50)	0.608±0.009 (+17.8%)	0.469±0.011 (+27.9%)	0.391±0.007 (+28.2%)	0.340±0.005 (+27.1%)	0.304±0.004 (+25.3%)	0.416±0.007 (+25.8%)
LRF-SLSVM (#LS=100)	0.646±0.010 (+25.4%)	0.495±0.009 (+34.9%)	0.411±0.007 (+34.8%)	0.356±0.006 (+33.2%)	0.317±0.005 (+30.9%)	0.438±0.007 (+32.4%)

The baseline algorithm is the regular query expansion algorithm (RF-QEX).

semantic categories than the latter. As a result, the relevance function based on the log data of users' relevance feedback will less accurately reflect similarity between two images, leading to the degradation in retrieval performance. Nevertheless, we still observe significant improvements with the 50-Category data set. The average improvement in MAP measure is 8.2 percent for the LRF-QEX algorithm over the RF-QEX algorithm, and 10.5 percent for the LRF-SVM algorithm over the RF-SVM algorithm.

Based on the above observations, we conclude that the algorithms for log-based relevance feedback can be expected to outperform the regular relevance feedback schemes.

6.5 Performance Evaluation on Small Log Data

In a real-world CBIR application, it may be difficult to collect a large amount of log data, particularly early in the life of a CBIR system. Hence, it is important to evaluate the performance of a log-based relevance feedback algorithm with a small amount of log data. To this end, we evaluate the compared schemes by varying the amount of log data. In particular, for each data set, only half of its log data is used for log-based relevance feedback. This amounts to 50 log sessions for the 20-Category data set, and 75 log sessions for the 50-Category data set. The empirical results for the reduced log data are shown in Table 2 and Table 3.

According to the two tables, we observe that the log-based relevance feedback algorithm by Soft Label SVM (LRF-SLSVM) achieves a promising improvement even with a limited amount of log data. Most impressively, the mean average precision (MAP) of Soft Label SVM using only half of the log sessions is better than the LRF-QEX approach that uses all the log sessions. For the 20-Category data set, with only 50 log sessions, the LRF-SLSVM algorithm outperforms the baseline algorithm (RF-QEX) by 25.8 percent and also enjoys a 6.9 percent improvement over the regular RF-SVM algorithm. The improvement on the 50-Category data set is again less than the 20-Category one. But, the LRF-SLSVM algorithm still outperforms the RF-QEX algorithm by 15.5 percent and has a 5.5 percent improvement over the RF-SVM algorithm with only 75 log sessions.

6.6 Performance Evaluation on Noisy Log Data

The presence of noise in the log data is unavoidable when the data is collected from a real-world CBIR application. It is therefore important to evaluate whether a good log-based relevance feedback algorithm is resilient to the noise present in the log data.

In this section, we conduct experiments to evaluate the robustness of algorithms on the log data with different levels of noise, meanwhile we compare the performance of SLSVM using different regularization strategies. Two sets of log data on both data sets, with different noise percentages, are employed to evaluate the algorithms. For each of the two data sets, two sets of log data are provided. The noise levels for the 20-Category data set are 7.8 percent and 16.2 percent, respectively, and 7.7 percent and 17.1 percent, respectively, for the 50-Category data set. In addition to varying the amount of noise in the log data, we also conduct experiments for the proposed algorithm LRF-SLSVM with different setup of the two weight parameters C_S and C_H. Two configurations of C_S and C_H are used in this experiment: $C_S = C_H$, which we refer to as $(\mathrm{LRF-SLSVM^{SR}})$, and $C_H > C_S$, which we refer to as $(\mathrm{LRF-SLSVM^{DR}})$.

Table 4 and Table 5 show the comparison results on the data sets with different noise percentages. As expected, performance of the algorithms degrades when a large amount of noise is present in the log data. Compared with other approaches, the Soft Label schemes are more tolerant to the noisy log data. Both the two Soft Label algorithms, i.e., $\mathrm{LRF\text{-}SLSVM^{SR}}$ and $\mathrm{LRF\text{-}SLSVM^{DR}}$, achieve better performance than the standard SVM algorithm. More impressively, we observe that the performance of the $\mathrm{LRF\text{-}SLSVM^{DR}}$ scheme with highly noisy log data is comparable to or better than that of the standard SVM using log data of low noise. Specifically, on the 20-Category data set, the standard SVM method $(\mathrm{LRF\text{-}SVM^{DR}})$ enjoys a 25.9 percent improvement in MAP over the baseline algorithm under the

TABLE 3
Performance Comparisons (Average Precision) on Different Amounts of Log Data on the 50-Category Data Set

Algorithm	TOP-20	TOP-40	TOP-60	TOP-80	TOP-100	MAP
RF-QEX (Baseline)	0.465±0.019	0.348±0.015	0.294±0.009	0.258±0.007	0.233±0.007	0.313±0.011
RF-SVM	0.489±0.010 (+5.2%)	0.386±0.006 (+11.0%)	0.323±0.006 (+9.7%)	0.282±0.004 (+9.4%)	0.254±0.004 (+8.8%)	0.343±0.006 (+9.5%)
LRF-QEX (#LS=75)	0.509±0.016 (+9.4%)	0.366±0.013 (+5.2%)	0.304±0.010 (+3.2%)	0.264±0.008 (+2.2%)	0.238±0.008 (+2.2%)	0.328±0.011 (+4.3%)
LRF-QEX (#LS=150)	0.543±0.017 (+16.8%)	0.380±0.015 (+9.4%)	0.313±0.011 (+6.4%)	0.271±0.010 (+5.0%)	0.243±0.010 (+4.3%)	0.342±0.012 (+8.2%)
LRF-SLSVM (#LS=75)	0.536±0.016 (+15.2%)	0.407±0.009 (+17.1%)	0.341±0.008 (+16.0%)	0.295±0.007 (+14.6%)	0.262±0.005 (+12.6%)	0.363±0.009 (+15.5%)
LRF-SLSVM (#LS=150)	0.568±0.020 (+22.0%)	0.429±0.013 (+23.3%)	0.357±0.011 (+21.4%)	0.308±0.008 (+19.4%)	0.272±0.007 (+16.7%)	0.381±0.011 (+21.0%)

The baseline algorithm is the regular query expansion algorithm (RF-QEX).

TABLE 4
Performance Comparisons (Average Precision) on the Log Data with Different Amounts of Noise on the 20-Category Data Set

Algorithm	TOP-20	TOP-40	TOP-60	TOP-80	TOP-100	MAP
RF-QEX (Baseline)	0.516±0.017	0.367±0.010	0.305±0.009	0.267±0.009	0.243±0.008	0.332±0.011
RF-SVM	0.535±0.006 (+3.8%)	0.433±0.002 (+18.1%)	0.370±0.001 (+21.2%)	0.325±0.002 (+21.7%)	0.292±0.002 (+20.4%)	0.387±0.002 (+17.8%)
LRF-SVM (Low Noise)	0.626±0.010 (+21.3%)	0.474±0.006 (+29.2%)	0.391±0.002 (+28.3%)	0.335±0.003 (+25.5%)	0.298±0.003 (+22.8%)	0.418±0.005 (+25.9%)
LRF-SLSVM$^{\mathrm{SR}}$ (Low Noise)	0.635±0.012 (+23.1%)	0.484±0.007 (+32.1%)	0.401±0.004 (+31.4%)	0.344±0.004 (+28.6%)	0.305±0.003 (+25.8%)	0.427±0.006 (+28.7%)
LRF-SLSVM$^{\mathrm{DR}}$ (Low Noise)	0.646±0.010 (+25.4%)	0.495±0.009 (+34.9%)	0.411±0.007 (+34.8%)	0.356±0.006 (+33.2%)	0.317±0.005 (+30.9%)	0.438±0.007 (+32.4%)
LRF-SVM (High Noise)	0.557±0.021 (+8.1%)	0.433±0.016 (+18.2%)	0.366±0.010 (+20.1%)	0.317±0.009 (+18.6%)	0.283±0.009 (+16.6%)	0.386±0.013 (+17.0%)
LRF-SLSVM$^{\mathrm{SR}}$ (High Noise)	0.584±0.010 (+13.3%)	0.451±0.005 (+23.1%)	0.378±0.002 (+24.1%)	0.327±0.004 (+22.5%)	0.293±0.004 (+20.7%)	0.401±0.005 (+21.3%)
LRF-SLSVM$^{\mathrm{DR}}$ (High Noise)	0.608±0.011 (+18.0%)	0.470±0.009 (+28.3%)	0.398±0.009 (+30.6%)	0.348±0.011 (+30.1%)	0.310±0.009 (+27.8%)	0.421±0.010 (+27.5%)

The baseline algorithm is the regular query expansion algorithm (RF-QEX).

low noisy log data, while the LRF-SLSVM$^{\mathrm{DR}}$ method achieves a 27.5 percent improvement even with the highly noisy log data. Similar results can also be observed on the 50-Category data set. Based on the above observation, we conclude empirically that the Soft Label SVM scheme is more tolerant to the noise than the standard SVM. Finally, comparing the two different configurations of LRF-SLSVM, we observe that LRF-SLSVM$^{\mathrm{DR}}$ performs slightly better than LRF-SLSVM$^{\mathrm{SR}}$ for both data sets. This is consistent with our hypothesis, i.e., it is more important to correctly classify the hard-labeled examples than the ones with soft labels.

6.7 Computational Complexity and Evaluation of Time Efficiency

Although we have observed significant improvement of our log-based relevance feedback scheme from the above experimental results, it is evident that our scheme requires extra computational cost compared with a regular relevance feedback scheme. Hence, it is necessary to analyze the computational complexity of the log-based relevance feedback scheme and empirically evaluate the time efficiency of our proposed scheme. In our log-based relevance feedback framework, there are two main components that contribute the most to the computational costs. One is the computation of the relevance function on the feedback log data, and the other is the learning of the relevance function on the low-

TABLE 5
Performance Comparisons (Average Precision) on the Log Data with Different Amounts of Noise on the 50-Category Data Set

Algorithm	TOP-20	TOP-40	TOP-60	TOP-80	TOP-100	MAP
RF-QEX (Baseline)	0.465±0.019	0.348±0.015	0.294±0.009	0.258±0.007	0.233±0.007	0.313±0.011
RF-SVM	0.489±0.010 (+5.2%)	0.386±0.006 (+11.0%)	0.323±0.006 (+9.7%)	0.282±0.004 (+9.4%)	0.254±0.004 (+8.8%)	0.343±0.006 (+9.5%)
LRF-SVM (Low Noise)	0.547±0.015 (+17.6%)	0.406±0.009 (+16.7%)	0.337±0.008 (+14.7%)	0.293±0.007 (+13.8%)	0.261±0.006 (+11.9%)	0.363±0.009 (+15.3%)
LRF-SLSVM$^{\mathrm{SR}}$ (Low Noise)	0.556±0.016 (+19.5%)	0.423±0.012 (+21.6%)	0.350±0.010 (+19.1%)	0.304±0.008 (+17.8%)	0.271±0.007 (+16.4%)	0.375±0.011 (+19.5%)
LRF-SLSVM$^{\mathrm{DR}}$ (Low Noise)	0.568±0.020 (+22.0%)	0.429±0.013 (+23.3%)	0.357±0.011 (+21.4%)	0.308±0.008 (+19.4%)	0.272±0.007 (+16.7%)	0.380±0.011 (+21.0%)
LRF-SVM (High Noise)	0.503±0.015 (+8.2%)	0.385±0.009 (+10.6%)	0.323±0.006 (+9.7%)	0.282±0.005 (+9.5%)	0.251±0.005 (+7.9%)	0.344±0.008 (+9.8%)
LRF-SLSVM$^{\mathrm{SR}}$ (High Noise)	0.519±0.018 (+11.6%)	0.393±0.012 (+12.9%)	0.328±0.010 (+11.4%)	0.289±0.008 (+12.2%)	0.257±0.007 (+10.2%)	0.352±0.011 (+12.2%)
LRF-SLSVM$^{\mathrm{DR}}$ (High Noise)	0.530±0.020 (+14.0%)	0.408±0.013 (+17.2%)	0.340±0.011 (+15.5%)	0.295±0.008 (+14.5%)	0.264±0.007 (+13.4%)	0.362±0.012 (+15.5%)

The baseline algorithm is the regular query expansion algorithm (RF-QEX).

level image features by the Soft Label SVM. It is straightforward to calculate the computational complexity for the former component, which is $\mathcal{O}(N_l \times N_{img} \times N_{log})$. Since N_l, i.e., the number of labeled images acquired from online user feedback, is regarded as a small constant, the time complexity in computing the log information is $\mathcal{O}(N_{img} \times N_{log})$. The major cost for the latter component is in training the SVM; this is determined by the implementation of the optimization problem in the SVM algorithms. In our experiments, the implementations of the SVM algorithms are based on the public libsvm library, for which more detailed analysis of computational cost can be found in [6]. Given that the computational cost for training SVM is highly dependent on the characteristics of the training examples, in the following, we will evaluate the efficiency of the proposed algorithm empirically.

To evaluate the time efficiency, we run 200 executions of relevance feedback with random queries, and record the time costs for both the RF-SVM algorithm and the LRF-SLSVM algorithm. Table 6 shows the experimental results of the time costs. The results indicate that extra time costs must be paid for running the LRF-SLSVM compared with the regular RF-SVM scheme. However, the results also suggest that the time costs of the LRF-SLSVM algorithm are still acceptable. For example, for the 50-Category data set with 150 log sessions, only 32.94 seconds are required for 200 relevance feedback executions, which amounts to only 0.165 seconds for each execution of feedback.

7 Limitation and Future Work

Based on the promising results achieved from the extensive evaluations, we can empirically conclude that our log-based relevance feedback scheme is an effective way to improve the traditional relevance feedback techniques by integrating log data of users' relevance feedback. Moreover, the Soft Label SVM algorithm has been demonstrated to be more resilient to the noise problem when solving the log-based relevance feedback problem. However, we must address the limitations of and the challenging issues with our scheme, as well as provide feasible directions for solving these problems in our future work.

The first limitation of our scheme may be the computational complexity problem. Two main computational costs are inherited. One is the relevance computing of log data; and the other is the training cost of Soft Label SVM. For the formal one, the computational cost can be critical when the number of log sessions are huge. Fortunately, our proposed incremental method in (5) can partially solve the problem. For the latter one, we can study more efficient decomposition techniques to solve our optimization problem, e.g., the parallel SVMs [11].

Second, it may be possible to learn the relevance function more effectively. In the current scheme, we only consider the classification model in the space of image features. It would be possible to apply the method in the reverse direction by first computing the soft labels from the image features and then building a classification model in the space of the users' relevance judgement. Furthermore, these

TABLE 6
Time Costs of the Proposed Schemes (Seconds)

Datasets	RF-SVM	LRF-SLSVM		
	$T_{\mathtt{SVM}}$	$T_{\mathtt{SLSVM}}$	T_{log}	T_{total}
20-Category	5.53	8.09	4.87	12.96
50-Category	13.14	16.85	16.10	32.94

two approaches can be integrated together through a cotraining algorithm [4].

Third, we realize that the selection of parameter C_H and C_S in the Soft Label SVM algorithm has a major impact on the final retrieval results when deploying the algorithm in the log-based relevance feedback problem. Although our empirical approach for choosing C_H and C_S has resulted in satisfactory performance, we plan to investigate other approaches in principle for tuning these two parameters effectively, e.g., the entire regularization path approach for studying the parameters [12].

Finally, the noise problem could be handled in other ways. For example, to alleviate the negative effect from noisy log data, we can modify the Soft Label SVM by enforcing an upper bound on the error terms in the optimization of the Soft Label SVM.

8 Conclusions

We have proposed a unified log-based relevance feedback framework for integrating log data of user feedback with regular relevance feedback for image retrieval. Our framework first computes the relevance function on the log data of user feedback and then combines the relevance information with regular relevance feedback for the retrieval task. In order to address the noisy log data problem in real-world applications, we propose a novel learning algorithm to solve the log-based relevance feedback problem. The proposed algorithm, named Soft Label Support Vector Machine, is based on the solid regularization theory. We have conducted an extensive set of experiments on a sophisticated testbed for evaluating the performance of a number of algorithms on our log-based relevance feedback scheme. The promising experimental results have confirmed that our proposed algorithms are effective in improving the performance of traditional relevance feedback in image retrieval.

The important contributions to the field in this work can be summarized as follows: First, we present a unified framework for studying the log-based relevance feedback problem. To the best of our knowledge, this work is among one of only a few pioneering investigations on incorporating both log data of users' feedback and online relevance feedback to improve image retrieval performance. Second, we propose a modified SVM algorithm, i.e., Soft Label SVM, to deal with the problem of noisy log data. Although we employ the Soft Label SVM only in the log-based relevance feedback problem, it can also be applied to other application areas, such as information filtering. Third, we have presented a comprehensive set of experimental procedures for evaluating image retrieval, and for examining various aspects of retrieval algorithms, including effectiveness, efficiency, robustness, and scalability.

Appendix

The Derivation for the Dual of OPT 2

Let us introduce the positive Lagrange multipliers $\alpha_i, i = 1, 2, \ldots, l+m$, one for each of the inequality constraints in the **OPT 2**, and μ_i for enforcing positivity of ξ_i. Then, the Lagrangian functional can be formulated as follows:

$$
\begin{aligned}
L(\mathbf{w}, \xi, b, \alpha, \mu) &= \frac{1}{2}\|\mathbf{w}\|^2 + C_H \sum_{i=1}^{l} \xi_i + C_S \sum_{i=l+1}^{l+m} y_i s_i \xi_i \\
&- \sum_{i=1}^{l} \alpha_i (y_i(\Phi(\mathbf{x}_i) \cdot \mathbf{w} - b) - 1 + \xi_i) \\
&- \sum_{i=l+1}^{l+m} \alpha_i (y_i(\Phi(\mathbf{x}_i) \cdot \mathbf{w} - b) - 1 + \xi_i) - \sum_{i=1}^{l+m} \mu_i \xi_i.
\end{aligned} \tag{17}
$$

By taking the partial derivative of L with respect to $\mathbf{w}, \xi_\mathbf{i}, b$, and ρ, we can obtain the following equations, respectively:

$$
\begin{aligned}
&\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{l+m} \alpha_i y_i \Phi(\mathbf{x}_i) = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{l+m} \alpha_i y_i \Phi(\mathbf{x}_i) \\
&\forall i = 1, \ldots, l \\
&\frac{\partial L}{\partial \xi_i} = C_H - \alpha_i - \mu_i = 0 \Rightarrow 0 \le \alpha_i \le C_H, \\
&\forall i = l+1, \ldots, l+m \\
&\frac{\partial L}{\partial \xi_i} = y_i s_i C_S - \alpha_i - \mu_i = 0 \Rightarrow 0 \le \alpha_i \le y_i s_i C_S, \\
&\frac{\partial L}{\partial b} = \sum_{i=1}^{l+m} \alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{l+m} \alpha_i y_i = 0.
\end{aligned}
$$

By substituting the above equations into (18), one can derive the dual of the original optimization problem as follows:

$$
\begin{aligned}
\max_{\alpha} \quad & \sum_{i=1}^{l+m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l+m} \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \\
\text{subject to} \quad & \sum_{i=1}^{l+m} \alpha_i y_i = 0 \\
& 0 \le \alpha_i \le C_H, i = 1, 2, \ldots, l, \\
& 0 \le \alpha_i \le y_i s_i C_S, i = l+1, l+2, \ldots, l+m.
\end{aligned}
$$

Acknowledgments

The work described in this paper was fully supported by two grants, one from the Shun Hing Institute of Advanced Engineering, and the other from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4205/04E).

References

[1] P. Anick, "Using Terminological Feedback for Web Search Refinement: A Log-Based Study," *Proc. 26th Ann. Int'l ACM SIGIR Conf.*, pp. 88-95, 2003.

[2] S. Berretti, A. Del Bimbo, and P. Pala, "Retrieval by Shape Similarity with Perceptual Distance and Effective Indexing," *IEEE Trans. Multimedia*, vol. 4, pp. 225-239, 2000.

[3] D. Blei and M.I. Jordan, "Modeling Annotated Data," *Proc. 26th Ann. Int'l ACM SIGIR Conf.*, pp. 127-134, 2003.

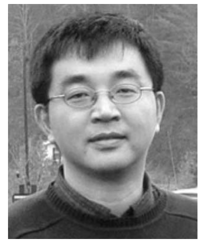
[4] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," *Proc. 11th Ann. Conf. Computational Learning Theory*, pp. 92-100, 1998.

[5] C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121-167, 1998.

[6] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.
[7] I.J. Cox, M. Miller, T. Minka, and P. Yianilos, "An Optimized Interaction Strategy for Bayesian Relevance Feedback," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 553-558, 1998.
[8] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, "Probabilistic Query Expansion Using Query Logs," *Proc. 11th Int'l Conf. World Wide Web,* pp. 325-332, 2002.
[9] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, "Query Expansion by Mining User Logs," *IEEE Trans. Knowledge and Data Eng.,* vol. 4, pp. 829-839, 2003.
[10] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization Networks and Support Vector Machines," *Advances in Computational Math.,* vol. 13, pp. 1-50, 2000.
[11] H.P. Graf, E. Cosatto, L. Bottou, I. Dourdanovic, and V. Vapnik, "Parallel Support Vector Machines: The Cascade SVM," *Advances in Neural Information Processing Systems,* 2005.
[12] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu, "The Entire Regularization Path for the Support Vector Machine," *J. Machine Learning Research,* vol. 5, pp. 1391-1415, 2004.
[13] X. He, O. King, W.-Y. Ma, M. Li, and H.J. Zhang, "Learning a Semantic Space from User's Relevance Feedback for Image Retrieval," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 13, no. 1, pp. 39-48, Jan. 2003.
[14] X. He, W.-Y. Ma, and H.-J. Zhang, "Learning an Image Manifold for Retrieval," *Proc. 12th ACM Int'l Conf. Multimedia,* pp. 17-23, 2004.
[15] C.H. Hoi and M.R. Lyu, "Biased Support Vector Machine for Relevance Feedback in Image Retrieval," *Proc. Int'l Joint Conf. Neural Networks,* pp. 3189-3194, 2004.
[16] C.H. Hoi and M.R. Lyu, "Group-Based Relevance Feeedback with Support Vector Machine Ensembles," *Proc. 17th Int'l Conf. Pattern Recognition,* pp. 874-877, 2004.
[17] C.H. Hoi and M.R. Lyu, "A Novel Log-Based Relevance Feedback Technique in Content-Based Image Retrieval," *Proc. 12th ACM Int'l Conf. Multimedia,* pp. 24-31, 2004.
[18] P. Hong, Q. Tian, and T.S. Huang, "Incorporate Support Vector Machines to Content-Based Image Retrieval with Relevant Feedback," *Proc. of IEEE Int'l Conf. Image Processing,* vol. 3, pp. 750-753, 2000.
[19] T.S. Huang and X.S. Zhou, "Image Retrieval by Relevance Feedback: From Heuristic Weight Adjustment to Optimal Learning Methods," *Proc. IEEE Int'l Conf. Image Processing,* vol. 3, pp. 2-5, Oct. 2001.
[20] Y. Ishikawa, R. Subramanya, and C. Faloutsos, "MindReader: Querying Databases through Multiple Examples," *Proc. 24th Int'l Conf. Very Large Data Bases,* pp. 218-227, 1998.
[21] A.K. Jain and A. Vailaya, "Shape-Based Retrieval: A Case Study with Trademark Image Database," *Pattern Recognition,* vol. 9, pp. 1369-1390, 1998.
[22] J. Jeon, V. Lavrenko, and R. Manmatha, "Automatic Image Annotation and Retrieval Using Cross-Media Relevance Models," *Proc. 26th Ann. Int'l ACM SIGIR Conf.,* pp. 119-126, 2003.
[23] T. Joachims, "Transductive Inference for Text Classification Using Support Vector Machines," *Proc. 16th Int'l Conf. Machine Learning,* pp. 200-209, 1999.
[24] J. Laaksonen, M. Koskela, and E. Oja, "Picsom: Self-Organizing Maps for Content-Based Image Retrieval," *Proc. Int'l Joint Conf. Neural Networks,* 1999.
[25] V. Lavrenko, R. Manmatha, and J. Jeon, "A Model for Learning the Semantics of Pictures," *Advances in Neural Information Processing Systems,* 2003.
[26] S. MacArthur, C. Brodley, and C. Shyu, "Relevance Feedback Decision Trees in Content-Based Image Retrieval," *Proc. IEEE Workshop Content-Based Access of Image and Video Libraries,* pp. 68-72, 2000.
[27] B. Manjunath, P. Wu, S. Newsam, and H. Shin, "A Texture Descriptor for Browsing and Similarity Retrieval," *J. Signal Processing: Image Comm.,* vol. 16, pp. 33-42, 2000.
[28] A.Y. Ng and M.I. Jordan, "On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes," *Advances in Neural Information Processing Systems,* vol. 14, pp. 841-848, 2001.
[29] J.C. Platt, "Fast Training of Support Vector Machines Using Sequential Minimal Optimization," *Advances in Kernel Methods—Support Vector Machines,* pp. 185-208, 1999.
[30] K. Porkaew, K. Chakrabarti, and S. Mehrotra, "Query Refinement for Multimedia Retrieval and Its Evaluation Techniques in MARS," *Proc. ACM Int'l Conf. Multimedia,* 1999.
[31] K. Porkaew, M. Ortega, and S. Mehrotra, "Query Reformulation for Content Based Multimedia Retrieval in MARS," *Proc. Int'l Conf. Multimedia Comm. Systems,* vol. 2, pp. 747-751, 1999.
[32] J. Rocchio, "Relevance Feedback in Information Retrieval," *The SMART Retrieval System: Experiments in Automatic Document Processing,* pp. 313-323, 1971.
[33] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra, "Relevance Feedback: A Power Tool in Interactive Content-Based Image Retrieval," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 8, no. 5, pp. 644-655, Sept. 1998.
[34] G. Salton and C. Buckley, "Improving Retrieval Performance by Relevance Feedback," *J. Am. Soc. for Information Science,* vol. 44, no. 4, pp. 288-287, 1990.
[35] G. Salton and M.J. McGill, *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.
[36] 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. Pattern Analysis and Machine Intelligence,* vol. 22, no. 12, pp. 1349-1380, Dec. 2000.
[37] J. Smith and S.-F. Chang, "Automated Image Retrieval Using Color and Texture," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 11, Nov. 1996.
[38] D. Tao and X. Tang, "Nonparametric Discriminant Analysis in Relevance Feedback for Content-Based Image Retrieval," *Proc. IEEE Int'l Conf. Pattern Recognition,* 2004.
[39] D. Tao and X. Tang, "Random Sampling Based SVM for Relevance Feedback Image Retrieval," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2004.
[40] K. Tieu and P. Viola, "Boosting Image Retrieval," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 228-235, 2000.
[41] S. Tong and E. Chang, "Support Vector Machine Active Learning for Image Retrieval," *Proc. Ninth ACM Int'l Conf. Multimedia,* pp. 107-118, 2001.
[42] V.N. Vapnik, *Statistical Learning Theory.* Wiley, 1998.
[43] N. Vasconcelos and A. Lippman, "Learning from User Feedback in Image Retrieval Systems," *Advances in Neural Information Processing Systems,* 1999.
[44] N. Vasconcelos and A. Lippman, "Bayesian Relevance Feedback for Content-Based Image Retrieval," *Proc. IEEE Workshop Content-Based Access of Image and Video Libraries,* pp. 63-67, 2000.
[45] Y. Wu, Q. Tian, and T.S. Huang, "Discriminant-Em Algorithm with Application to Image Retrieval," *IEEE Conf. Computer Vision and Pattern Recognition,* 2000.
[46] J. Xu and W.B. Croft, "Query Expansion Using Local and Global Document Analysis," *Proc. 19th Ann. Int'l ACM SIGIR Conf.,* pp. 4-11, 1996.
[47] L. Zhang, F. Lin, and B. Zhang, "Support Vector Machine Learning for Image Retrieval," *Proc. Int'l Conf. Image Processing,* vol. 2, pp. 721-724, 2001.
[48] T. Zhang and F.J. Oles, "A Probability Analysis on the Value of Unlabeled Data for Classification Problems," *Proc. 17th Int'l Conf. Machine Learning,* 2000.
[49] X.-D. Zhou, L. Zhang, L. Liu, Q. Zhang, and B.-L. Shi, "A Relevance Feedback Method in Image Retrieval by Analyzing Feedback Log File," *Proc. Int'l Conf. Machine Learning and Cybernetics,* vol. 3, pp. 1641-1646, 2002.
[50] X.S. Zhou and T.S. Huang, "Small Sample Learning During Multimedia Retrieval Using Biasmap," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2001.

Steven C.H. Hoi received the bachelor's degree in computer science and technology from Tsinghua University, Beijing, P.R. China, in 2002, and the Master's of Philosophy degree in computer science and engineering from Chinese University of Hong Kong, in 2004. He is currently a PhD candidate in the same department at the Chinese University of Hong Kong. His research interests are in multimedia information retrieval, pattern recognition, and statistical machine learning. He is a student member of the IEEE.

Michael R. Lyu received the BS degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1981, the MS degree in computer engineering from the University of California, Santa Barbara, in 1985, and the PhD degree in computer science from the University of California, Los Angeles, in 1988. He is currently a professor in the Department of Computer Science and Engineering at the Chinese University of Hong Kong. He is also director of the Video over InternEt and Wireless (VIEW) Technologies Laboratory. He was with the Jet Propulsion Laboratory as a technical staff member from 1988 to 1990. From 1990 to 1992, he was with the Department of Electrical and Computer Engineering at The University of Iowa, Iowa City, as an assistant professor. From 1992 to 1995, he was a member of the technical staff in the applied research area of Bell Communications Research (Bellcore), Morristown, New Jersey. From 1995 to 1997, he was a research member of the technical staff at Bell Laboratories, Murray Hill, New Jersey. His research interests include software reliability engineering, distributed systems, fault-tolerant computing, mobile networks, Web technologies, multimedia information processing, and E-commerce systems. He has published more than 200 refereed journal and conference papers in these areas. He received Best Paper Awards at ISSRE '98 and ISSRE '03. He has participated in more than 30 industrial projects and helped to develop many commercial systems and software tools. He was the editor of two book volumes: *Software Fault Tolerance* (New York: Wiley, 1995) and *The Handbook of Software Reliability Engineering* (Piscataway, NJ: IEEE and New York: McGraw-Hill, 1996). Professor Lyu initiated the First International Symposium on Software Reliability Engineering (ISSRE) in 1990. He was the program chair for ISSRE '96 and general chair for ISSRE '01. He was also PRDC'99 program cochair, WWW10 program cochair, SRDS'2005 program cochair, and PRDC'2005 general cochair, and served on program committees for many other conferences including HASE, ICECCS, ISIT, FTCS, DSN, ICDSN, EUROMICRO, APSEC, PRDC, PSAM, ICCCN, ISESE, and WI. He has been frequently invited as a keynote or tutorial speaker to conferences and workshops in the US, Europe, and Asia. He served on the editorial boards of *IEEE Transactions on Knowledge and Data Engineering* and *IEEE Transactions on Reliability*. He is currently an editor for the *Journal on Information Science and Engineering*. He is an IEEE fellow.



Rong Jin received the PhD degree in computer science from Carnegie Mellon University, 2003. He is assistant professor of computer science and engineering at Michigan State University. His research interests are in statistical machine learning and its application to information retrieval. He is a member of the IEEE.