Learning a Semantic Space From User's Relevance Feedback for Image Retrieval

Xiaofei He, Oliver King, Wei-Ying Ma, Mingjing Li, and Hong-Jiang Zhang, *Senior Member, IEEE*

***Abstract*—As current methods for content-based retrieval are incapable of capturing the semantics of images, we experiment with using spectral methods to infer a semantic space from user's relevance feedback, so that our system will gradually improve its retrieval performance through accumulated user interactions. In addition to the long-term learning process, we also model the traditional approaches to query refinement using relevance feedback as a short-term learning process. The proposed short- and long-term learning frameworks have been integrated into an image retrieval system. Experimental results on a large collection of images have shown the effectiveness and robustness of our proposed algorithms.**

***Index Terms*—Image retrieval, learning, semantics, singular value decomposition, user's relevance feedback.**

I. Introduction

DUE TO the rapid growth of the number of digital images, there is an increasing demand for effective image management tools. Conventional content-based image retrieval (CBIR) systems [6], [13], [17] use low-level features (color, texture, shape, etc.) automatically extracted from the images themselves to search for images relevant to a user's query. While there are research efforts to improve performance by using different low-level features, and by modifying the similarity measures constructed from them, it is argued in [19] that, as unconstrained object recognition is still beyond the reach of current technology, these content-based systems can at best capture only pre-attentive similarity, not semantic similarity.

In recent years, much has been written about relevance feedback in content-based image retrieval from the perspective of machine learning [20], [21], [23]–[25], yet most learning methods take into account only the current query session, while the knowledge obtained from the past user interactions with the system is forgotten. To compare the effects of different learning techniques, a useful distinction can be made between *short-term* learning within a single query session and *long-term* learning over the course of many query sessions. Short-term learning is memoryless and aims to improve the retrieval performance of the current query session. Long-term learning aims to accumulate knowledge from users, which could result in new feature representations for images in the database so that the system's future retrieval performance is enhanced. Both short- and long-term learning processes are useful in an image retrieval system, though the former has been the primary focus of research so far.

Despite much work on relevance feedback for image retrieval (i.e., short-term learning) in the past few years, little work has been done from the theoretical perspective. In contrast, computational on-line learning algorithms [8] have been well analyzed in text retrieval [2], [5], [10], [15]. These techniques have been better understood from a theoretical standpoint, leading to performance guarantees and guidance in parameter settings. In this paper, we use mistake-driven on-line learning algorithms to model the process of image retrieval based on user's relevance feedback. The on-line learning algorithm winnow [11] is used to train an image classifier for searching for more relevant images from the database based on the positive and negative examples provided by a user. Following the theoretical analysis in [11], we derive a mistake upper bound, i.e., a bound on how many relevance feedbacks are needed for reaching a satisfactory performance in image retrieval.

To address the limitations of current systems with regard to searching for images at the semantic level, we propose a long-term learning method that creates a semantic space implicitly, based on user interactions in a relevance feedback driven query-by-example system. The idea is that, after several rounds of relevance feedback, the user has a pool of images that are relevant to his query. Assuming these images belong to a semantic class, by aggregating such results we may incrementally construct a semantic space, with a concomitant improvement in the system's performance. We use the singular value decomposition (SVD) to reduce the dimensionality of the semantic space, both for savings in storage and for possible improvement in retrieval performance. Due to the dimensionality reduction, the relevant and irrelevant images in the semantic space may be no longer linearly separable. In this case, systems such as support vector machines (SVMs) can be used to learn the target function for retrieving relevant images. Our experiments show that the SVD helps to correlate relevance feedbacks from different search sessions and reduce the subjectivity and noise introduced by individual users.

The rest of this paper is organized as follows. Section II relates a list of previous works to our work and summarizes our contribution. Section III describes the proposed method for long-term learning. Section IV describes the proposed method for short-term learning, with theoretical analysis. Our *MiAlbum* image retrieval system is introduced in Section V. The experimental results are shown in Section VI. Finally, we give concluding remarks and discuss future work in Section VII.

Manuscript January 15, 2002; revised September 23, 2002. This paper was recommended by Associate Editor J. R. Smith.

X. He is with the Computer Science Department, University of Chicago, Chicago, IL 60637 USA (e-mail: xiaofei@cs.uchicago.edu).

O. King is with Harvard Medical School, Boston, MA 02115 USA (e-mail: ok@csua.berkeley.edu).

W.-Y. Ma is with Microsoft Research Asia, Beijing, 100080 China (e-mail: wyma@microsoft.com).

M. Li and H.-J. Zhang are with Microsoft Research Asia, Beijing 100080 China (e-mail: mjli@microsoft.com; hjzhang@microsoft.com).

Digital Object Identifier 10.1109/TCSVT.2002.808087

II. Previous Work

One of the most popular models used in information retrieval is the vector space model [18]. Various retrieval techniques have been developed for this model, including the method of relevance feedback. Most previous research on relevance feedback has fallen into the following three categories: retrieval based on query point movement [17], retrieval based on re-weighting of different feature dimensions [7], and retrieval based on updating the probability distribution of images in the database [4].

In recent years, some learning-based approaches have been proposed. Wu *et al.* [24] proposed a Discriminant-EM algorithm within the transductive learning framework in which both labeled and unlabeled images are used. Tieu *et al.* [22] presented a framework for image retrieval based on representing images with a very large set of highly selective features. Queries are interactively learned online with a simple boosting algorithm. Tong *et al.* [23] proposed the use of a SVM active learning algorithm for conducting effective relevance feedback for image retrieval. While most machine learning algorithms are passive in the sense that they are generally applied using a randomly selected training set, the SVM active learning algorithm chooses the most informative images within the database, and asks the user to label these.

All of these approaches have achieved good empirical results. However, a common limitation of them is that they do not have a mechanism to memorize or accumulate relevance feedback information provided by users; consequently, the knowledge obtained from the previous queries and relevance feedback is forgotten.

Cox *et al.* [3] showed that query-by-example performance may improve by placing images in a semantic space, even if the user does not actually query by keyword (i.e., if the semantic attributes inducing the similarity measure are hidden). In that experiment, pictures were visually examined to see which of approximately 125 keywords were relevant, and these ratings were used to construct a semantic space for the images.

In [9], an image retrieval system based on an information embedding scheme is proposed. Using relevance feedback, the system gradually embeds correlations between images from a high-level semantic perspective. The semantic relationships between images are captured and embedded into the system by splitting/merging image clusters and updating the correlation matrix. In this way, the user-provided information is gradually embedded into the system; however, the system may take a long time to converge, and may not converge to an optimal state.

Here, we summarize the novel contributions of our work.

1) A long-term learning method is proposed to infer a semantic space for improving the system's retrieval performance over time. It consists of two parts: learning semantics from user interactions and from image content. A technique based on SVD is proposed, to form a compact semantic feature representation and reduce the subjectivity and noise from an individual user.
2) An on-line learning model for the traditional relevance feedback methods for image retrieval is proposed. Based on the model, a theoretical analysis of at most how many feedbacks are needed is performed. We also show that the semantic space may no longer be linearly separable after dimensionality reduction. In this case, a SVM training algorithm is used to retrieve relevant images from the database.
3) An image retrieval system integrating both the short- and long-term learning algorithms is developed. Our experimental results demonstrate that the proposed learning techniques are effective in capturing user's relevance feedback for improving the system's short- and long-term performances.

III. Long-Term Learning: Inferring a Semantic Space

Most existing relevance feedback techniques focus on improving the retrieval performance of the current query session, and the knowledge obtained from past user interactions with the system is forgotten. In this section, we describe a long-term learning approach for constructing a semantic space from user interactions and image content. The proposed learning technique is able to accumulate knowledge from users over time, and gradually enhance the retrieval performance of the system.

A. Hidden Semantic Features

We adopt the vector space model of information retrieval [18] to represent the semantic space constructed from user-and-system interactions. In this model, one has a matrix B (say of size $m \times n$), whose rows correspond to images and whose columns correspond to attributes. In a traditional image retrieval system, these columns correspond to low-level features (e.g., color and texture) or pre-annotated high-level semantic attributes (e.g., dog, cat, tree, people, etc). B_{ij} is a measure of the extent to which image i has attribute j; it may be binary, weighted by frequency, etc. The ith row of B may then be regarded as the coordinates of the ith image in an n-dimensional vector space, and the dot-product between rows i_1 and i_2 of B may be regarded as a measure of the similarity between images i_1 and i_2. Dividing this dot-product by the norms of the rows i_1 and i_2 gives the cosine of the angle between rows i_1 and i_2, another commonly used similarity measure.

We argue that the images marked by the user as positive examples in a query session often share a common semantic attribute. Since we do not know the exact meaning of the attribute unless the user specifically provides such information, we call it a *hidden semantic feature*. The hidden semantic features accumulated from user-and-system interactions can be used to infer a semantic space B for image retrieval. We discuss how to construct such a space in the following.

B. Constructing a Semantic Space

Let us assume that there exists a semantic matrix B for a database of m images. A row vector (n-dimensional) of the matrix B represents the hidden semantic features of an image. A query $\boldsymbol{q}$ may, like the image, be represented as an n-dimensional vector, and the retrieval results $\boldsymbol{r}$ of the query as an m-dimensional vector, with $\boldsymbol{r}(i)$ the similarity of $\boldsymbol{q}$ to row i of B. Concisely, $B\boldsymbol{q} = \boldsymbol{r}$, as illustrated in Fig. 1.

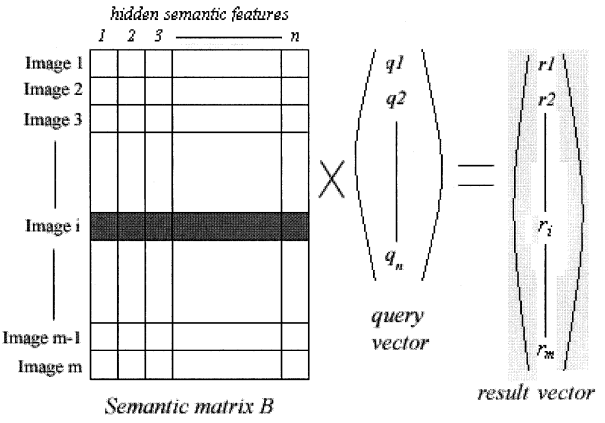

Fig. 1. Image retrieval in the semantic space can be thought of a matrix operation. B is a semantic matrix. $\boldsymbol{q}$ is a query vector. $B\boldsymbol{q} = \boldsymbol{r}$ is the result vector containing the similarity measure with each image in the database.

Learning Semantics From User Interactions: The long-term learning is essentially the process of inferring a semantic space B through knowledge of the result vectors $\boldsymbol{r}$ accumulated from the users' relevance feedback. Suppose that $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_k$ are the results of k queries with relevance feedback, with $\boldsymbol{r}_j(i) = 1$ if the ith image was deemed relevant to the jth query and with $\boldsymbol{r}_j(i) = 0$ otherwise. We seek a matrix B (whose rows represent the images) and query vectors $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k$ so that $B\boldsymbol{q}_j = \boldsymbol{r}_j$ for $j = 1, 2, \ldots, k$. Equivalently, defining R to be the $m \times k$ matrix whose jth column is $\boldsymbol{r}_j$, we seek an $m \times n$ matrix B and an $n \times k$ matrix Q such that $BQ = R$. Note that m, the number of images, is forced on us, as is k, but there is some choice in n. One possibility is to take $n = k$, $B = R$, and $Q = I$. In this solution, the jth column of Q, which stands for the jth query session, has jth entry one, and all other entries zero. Hence, this query session is going to retrieve those images having the jth hidden semantic feature. Multiplying B by $\boldsymbol{q}_j$, we get $\boldsymbol{r}_j$, which is the retrieval result. Fig. 2 shows a simple example of a semantic space constructed after three query sessions.

Reduce Semantic Space Using SVD: In the above section, choosing $Q = I$ implies that all queries are orthogonal. But, in practice, different queries may involve common high-level semantic features. Simply appending each retrieval result $\boldsymbol{r}$ as a column vector in matrix B does not exploit the correlation between queries. Another consequence is that the size of B grows linearly as the number of query sessions increases.

For storage and performance improvements, it is desirable to merge related hidden semantic features and construct a lower dimensional space B. We may compute the SVD of R, which expresses $R = USV^T$, with $U^TU = I$, $V^TV = I$, and S diagonal. Note that the column vectors of U and V are eigenvectors of RR^T and R^TR, respectively. Let p be the rank of R (which is equal to the number of nonzero entries on the diagonal of S). It can be at most $\min(m, k)$, and is possibly much smaller, since there may be linear dependencies among the $\boldsymbol{r}_j$ (for instance, when one semantic category is the disjoint union of others).

If we delete all but the first p columns of U and V, and all but the upper $p \times p$ submatrix of S, then we still have $R = USV^T$. Thus, we can let B be the $m \times p$ matrix US, and Q the $p \times k$ matrix V^T. Since $RR^T = (BV^T)(BV^T)^T = B(V^TV)B^T = BB^T$, this gives the same inter-image similarities as using $B = R$, but with reduced storage if $p < k$. (R is usually quite sparse, while US may not be.) Note that the "queries" (i.e., the columns of Q) will no longer in general be orthogonal if $p < k$.

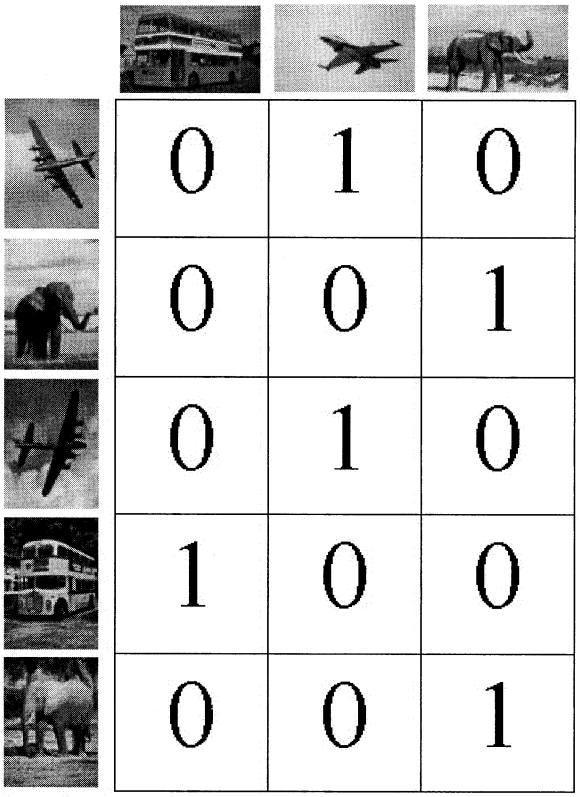

Fig. 2. Simple example of a semantic space constructed after three query sessions. There are five images in the database. The images on the top are query images. For each image in the database, its corresponding entry of the semantic matrix is set to be 1 if it is marked as relevant.

Now our result vectors $\boldsymbol{r}_i$ are constructed from user judgments as to which images are relevant to a given query (i.e., which images contain the same hidden semantic feature). But as the user does not generally inspect all the images in the database, there may be some spurious zeros in $\boldsymbol{r}$, and different users may disagree on certain images, even if seeking essentially the same semantic class. Thus, the matrix R may be noisy and of artificially increased rank. The cleaner, ideal results may be generated by a linear process of rank less than rank(R). By taking $n <$ rank(R), deleting all but the first n columns of U and V and all but the upper $n \times n$ submatrix of S as before, and by taking $B = US$ and $Q = V^T$ as before, we obtain a still lower dimensional semantic space. It is no longer true that $BQ = R$, but instead BQ is the best rank-n approximation of R, in the least squares sense (i.e., under the Frobenius norm).

The above idea is similar to latent semantic indexing (LSI) for text retrieval, and it has been shown that relative precision can improve by 30% by reducing the rank of document-term matrices in this fashion [1]. The claim is that meaningless distinctions between words are reduced. Theoretical results that go some ways toward explaining these empirical successes appear in [14], though under fairly restrictive hypotheses.

C. Updating the Semantic Space

In a real-world implementation of the retrieval system, the semantic matrix B may be periodically replaced by US in a reduced-rank SVD approximation USV^T of B. If one does not have an *a priori* estimate of the rank of the underlying linear process, one may resort to *ad hoc* methods for choosing the dimension for the reduced rank SVD based on examining the sizes of the singular values, or assessing the retrieval performance of the algorithm. In theory, the optimal rank is closely related to the number of semantic categories in the image database. If this number can be roughly estimated, it can be used as a guideline to select the best rank for updating the semantic space. As more vectors $\boldsymbol{r}$ are appended and B becomes bloated, B is then subjected to an SVD again to keep its rank within a certain range.

Note that, in a real-world application, we do not need to compute the SVD of B (or R) explicitly. We only need to compute the matrix V of the eigenvectors of B^TB. The semantic matrix is replaced by $BV(=US)$. The matrix V is called the *transformation matrix*. Since the semantic space is periodically reduced to a low-dimensional space, the dimension of the matrix B^TB is usually not high.

IV. Short-Term Learning: Learning a Classifier From Examples

In Section III, we described our algorithm for constructing a semantic space. With this semantic space, the aim of short-term learning is to infer the user's information need by applying supervised learning to build a classifier for differentiating semantically relevant images in the database from irrelevant ones. In this section, we first introduce the idea of the target function corresponding to a user's query.

A. Target Function in the Semantic Space

Our proposed short-term learning for image retrieval can be modeled as the following process: learn a function $g(\mathbf{x})$ which takes an image ($\mathbf{x}$ represents its n-dimensional feature vector) as input, and outputs 1 if this image is relevant and outputs 0 if it is irrelevant. Hence, the system uses $g(\mathbf{x})$ to distinguish relevant images from irrelevant ones. The goal of the short-term learning is to learn $g(\mathbf{x})$ and to make as few mistakes as possible, assuming that both the choice of relevant features and the choice of feedback examples are under the control of the user. Here, *relevant features* are those hidden semantic features that the user desires. We call $g(\mathbf{x})$ the *target function*. In other words, the goal is to train a classifier to label each image within the database, such that the classifier's labeling agrees with the user's labeling for all images.

B. Representing Query Example With Hidden Semantics

When an example image is presented to the system as a query, its low-level features (color, texture, etc) are extracted to conduct the first iteration of the search. Note that we do not have hidden semantic features for images unless they are in the database. After the first retrieval, the semantic representation of the query image can be formed based on the user's relevance feedback as follows: suppose the user marks s positive examples and t negative examples from among the first batch of retrieved images. Each of these $s+t$ images is represented by a semantic vector $\mathbf{x}^j$, with $j = 1, \ldots, s$ for the positive examples and $j = s+1, \ldots, s+t$ for the negative examples. Then the semantic feature for the query image can be represented as

$$\boldsymbol{q} = (q_1, q_2, \ldots, q_n)$$

where

$$q_i = (x_i^1 \vee x_i^2 \vee \cdots \vee x_i^s) \wedge \overline{(x_i^{s+1} \vee x_i^{s+2} \vee \cdots \vee x_i^{s+t})}, \quad i = 1, 2, \ldots, n$$

and where x_i^j is the ith element of the semantic vector $\mathbf{x}^j$. In the semantic space after SVD dimensionality reduction, the semantic vectors are no longer Boolean. In this case, we simply use a linear combination of the relevant images to represent the query image, as follows:

$$q_i = \frac{1}{s}\sum_{j=1}^{s} x_i^j.$$

C. Learning a Classifier in the Boolean Semantic Space

An image in n-dimensional Boolean semantic space is represented by a Boolean vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. The image has the ith hidden semantic feature if and only if $x_i = 1$. Thus, the task is to learn a target (discriminating) function g: $\{0, 1\}^n \rightarrow \{0, 1\}$. If the output is one, the system classifies the image as relevant, while if the output is zero, the system classifies the image as irrelevant. We assume in this analysis that users seek images having any one of some subset of relevant hidden semantic features. Then the optimal g is a disjunction function $g^{\text{opt}}(\mathbf{x}) = x_{i_1} \vee x_{i_2} \vee \cdots \vee x_{i_k}$, where $i_1, i_2, \ldots, i_k$ are the subscripts of the k relevant features, i.e., those hidden semantic features which the user desires. We also assume that the user acts as the disjunction function $g^{\text{opt}}(\mathbf{x})$ to teach the search engine. That is, for a given image, the user classifies it as positive example if it has at least one relevant feature. Otherwise, the user classifies it as a negative example. Since the images classified by $g^{\text{opt}}(\mathbf{x})$ are linearly separable (note that for any concept $x_{i_1} \vee x_{i_2} \vee \cdots \vee x_{i_k}$, a separating hyperplane is given by $x_{i_1} + x_{i_2} + \cdots + x_{i_k} \geq 0.5$), our goal is to find a linear hyperplane that separates the images with at least one relevant feature from those images with no relevant feature, as $g^{\text{opt}}(\mathbf{x})$ does. In our system, the linear discriminant function is defined as follows:

$$g(\mathbf{x}) = \begin{cases} 1, & \text{if } h_{\text{score}}(\mathbf{x}) \geq \theta \\ 0, & \text{if } h_{\text{score}}(\mathbf{x}) < \theta \end{cases}$$

where $h_{\text{score}}(\mathbf{x})$ is a function to evaluate the score of image $\mathbf{x}$ while ranking and θ is a threshold. The simplest score functions are linear; that is, they may be expressed as the dot product of a weight vector $\mathbf{w}$ and the hidden semantic feature vector $\mathbf{x}$ as follows:

$$h_{\text{score}}(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) = \sum_{j=1}^{n} w_j x_j.$$

Mistake-Driven Learning: Traditionally, the user's relevance feedbacks are used to update the query vector or adjust the weighting of different dimensions. This process can be

viewed as an on-line learning process in which the image retrieval system acts as a learner and the user acts as a teacher. The typical retrieval process is outlined as follows.

1) The user provides his relevance feedback to the system by labeling images as "relevant" or "irrelevant."
2) The system compares the user's judgment with the one generated by the current target function $g(\mathbf{x})$.
3) The system modifies $g(\mathbf{x})$ such that it generates a judgment coherent with the user's feedback.

If the system's judgment disagrees with that of the user, we say that the system makes a mistake. A mistake-driven learning algorithm updates $g(\mathbf{x})$ only when a mistake is made. In this section, we use a variant of Littlestone's winnow [11], one of the most widely used on-line learning algorithms for linear functions, to perform short-term learning for image retrieval.

Winnow-Like Learning Algorithm: With user's relevance feedback, our algorithm can learn the disjunction of hidden semantic features that the user desires.

A winnow-like mistake-driven on-line learning algorithm is used to learn the discriminant function $g(\mathbf{x})$. Initially, the weight vector $\mathbf{w}$ is set to be the query vector $\mathbf{q}$, which is obtained by the method described in Section IV-B. Those images with the highest scores, along with some random images, are presented to the user. If the current classifier labels an image $\mathbf{x}$ as "irrelevant" ([i.e., if $h_{score}(\mathbf{x}) \leq \theta$) while the user labels $\mathbf{x}$ as "relevant," we say a *positive mistake* occurs. Similarly, if the current classifier labels image $\mathbf{x}$ as "relevant" (i.e., if $h_{score}(\mathbf{x}) > \theta$) while the user labels $\mathbf{x}$ as "irrelevant," we say a *negative mistake* occurs. When the user's relevance feedback contradicts the current classification, the algorithm updates the weight vector as follows:

- Negative mistake:

$$w_i^{(t+1)} = \begin{cases} \dfrac{w_i^{(t)}}{\alpha}, & \text{if } x_i = 1 \\ w_i^{(t)}, & \text{if } x_i = 0 \end{cases}$$

- Positive mistake:

$$w_i^{(t+1)} = \begin{cases} w_i^{(t)}, & \text{if } x_i = 0 \\ 1, & \text{if } x_i = 1 \text{ and } w_i^{(t)} = 0 \\ \alpha w_i^{(t)}, & \text{if } x_i = 1 \text{ and } w_i^{(t)} \neq 0 \end{cases}$$

where α controls the adjustment rate and is greater than one.

How Many Feedbacks are Needed at Most—Theoretical Analysis of Mistake Bound: Despite tremendous research on using relevance feedback for image retrieval, little theoretical analysis has been performed so far. In this section, we provide a theoretical analysis of the mistake upper bound for the winnow-like algorithm. We regard each query as a classification problem, and train a linear classifier to discriminate between relevant and irrelevant images in the database. A linear classifier is represented by a pair $(\mathbf{w}, \theta)$, where $\mathbf{w} \in R^n$ is an n-dimensional weight vector and $\theta \in R$ is a threshold.

During the user interaction, the algorithm updates the weight vector each time a mistake occurs. Our goal is to minimize the total number of mistakes that the algorithm makes, so that the user can retrieve the target images as quickly as possible. The following theorem gives a theoretical upper bound on the required number of feedbacks.

Theorem 1: Assume n is the total number of hidden semantic features in the database. The winnow-like image retrieval algorithm with threshold θ and adjusting rate α learns the class of disjunctions over the n-dimensional Boolean vector space in the mistake-bound model, making at most $E = \alpha n/((\alpha-1)\theta) + [\alpha n/((\alpha-1)\theta)+\alpha+1]k(1+\log_\alpha \theta)$ mistakes when the target concept is a disjunction of k hidden semantic features.

Littlestone proved a similar result in [11]; since we use a slightly different update rule, we give a sketch of the proof in the Appendix. This gives us an estimate of at most how many feedbacks are needed. It should be pointed out, however, that this result is obtained in an idealized setting. In the real world, the user is not an optimal teacher, in most cases. That is, sometimes the user is unable to tell whether an image is relevant or irrelevant. Estimating the mistake-bound under such conditions is beyond the scope of this paper and is left for future studies.

D. Learning a Classifier in the Dimensionality-Reduced Semantic Space

In low-level feature (color, texture, shape, etc.) space, or in semantic space after dimensionality reduction, the representation of an image is no longer a Boolean vector, but a real-valued vector. Also, the relevant images and irrelevant images [determined by $g^{\text{opt}}(\mathbf{x})$] may no longer be linearly separable in the dimensionality-reduced semantic space.

In the Boolean semantic space, a linear classifier is given by a pair $(\mathbf{w}, \theta)$, where $\mathbf{w} \in R^n$ is an n-dimensional weight vector and $\theta \in R$ is a threshold. To be consistent with the previous section, we will still use $\mathbf{w}$ to denote a weight vector, but without loss of generality we can assume that the threshold is zero, by making the following modifications.

- Append a new dimension to $\mathbf{w}$ with value of $-\theta$

$$\mathbf{w}' \leftarrow (\mathbf{w}, -\theta).$$

- Append a new dimension to $\mathbf{x}$ with value of 1

$$\mathbf{x}' \leftarrow (\mathbf{x}, 1).$$

- Append a new orthogonal column vector to the transformation matrix V (see Section III)

$$V' \leftarrow \begin{bmatrix} V & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}.$$

(Here, $\mathbf{0}$ is a column vector taking 0 at each entry.)

Consequently, the transformed image vector in the reduced-dimension semantic space is also appended a new dimension with value of 1

$$\mathbf{y}' \leftarrow \mathbf{x}'V' = (\mathbf{x}V, 1) = (\mathbf{y}, 1).$$

Hence, the linear classifier can be represented by a single weight vector $\mathbf{w}$, which is called a linear *separator*.

Let S_w denote the set of all the linear separators in the original semantic space and let S_V denote the subspace spanned by the column vectors of matrix V. We have the following theorem, which is proven in the Appendix.

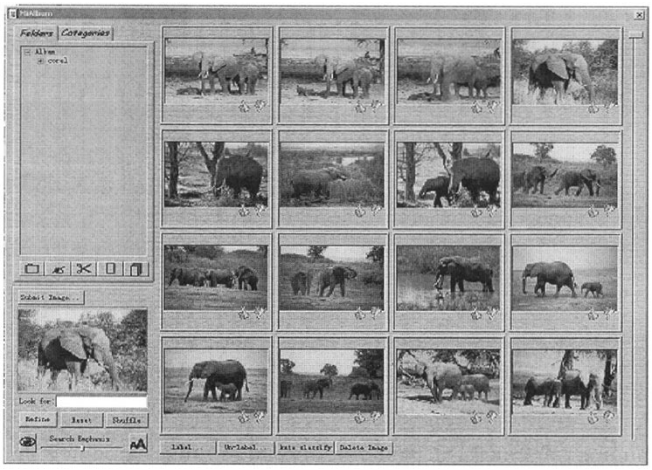

Fig. 3. MiAlbum image retrieval system. The user can provide his feedback by clicking "thumb up" or "thumb down" buttons associated with each retrieved image.

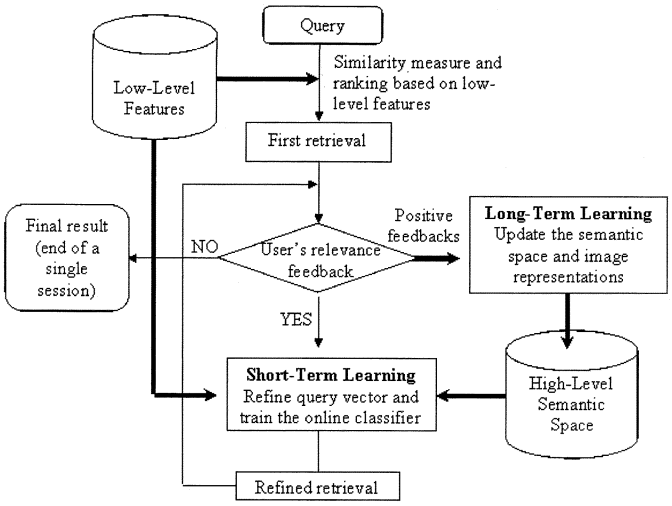

Fig. 4. Design of our system, which is equipped with both short- and long-term learning capabilities.

Theorem 2: The relevant images and irrelevant images are linearly separable in the reduced-dimension semantic space if and only if $S_{\mathbf{w}} \cap S_V \neq \phi$.

As the theorem indicates, if S_w and S_V are disjoint, then there does not exist a hyperplane that separates the relevant images from the irrelevant images. In this case, a SVM training algorithm can be used to learn a nonlinear target function for retrieving relevant images.

Another motivation for using an SVM is the small sample size issue in image retrieval. The number of training examples fed back by the user is usually small (six per round of interaction in our experiment) relative to the dimension of the feature space (from dozens to hundreds, or even more), while the number of semantic classes is large for most real-world image databases. SVMs make no assumptions on the distribution of the data and can, therefore, be applied even when we do not have enough knowledge to estimate the distribution that produced the input data.

SVMs: SVMs are a family of pattern classification algorithms developed by Vapnik [22] and collaborators. SVM training algorithms are based on the idea of *structural risk minimization* rather than *empirical risk minimization*, and give rise to new ways of training polynomial, neural network, and radial basis function (RBF) classifiers.

We shall consider SVMs in the binary classification setting. We assume that we have a data set $D = \{\mathbf{x}_i, y_i\}_{i=1}^{t}$ of labeled examples, where $y_i \in \{-1, 1\}$, and we wish to select, among the infinite number of linear classifiers that separate the data, one that minimizes the generalization error, or at least minimizes an upper bound on it. In [22], it is shown that the hyperplane with this property is the one that leaves the maximum margin between the two classes. Given a new data point $\mathbf{x}$ to classify, a label is assigned according to its relationship to the decision boundary, and the corresponding decision function is

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{t} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x}\rangle - b\right).$$

From this equation, it is possible to see that the α_i associated with the training point $\mathbf{x}_i$ expresses the strength with which that point is embedded in the final decision function. A remarkable property of this alternative representation is that often only a subset of the points will be associated with nonzero α_i. These points are called *support vectors* and are the points that lie closest to the separating hyperplane.

The nonlinear SVM implicitly maps the input variable into a high-dimensional (often infinite-dimensional) space, and applies the linear SVM in the space. Computationally, this can be achieved by the application of a (reproducing) kernel. The corresponding nonlinear decision function is

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{t} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) - b\right)$$

where K is the kernel function. Some typical kernel functions include polynomial kernels, Gaussian RBF kernels, and sigmoid kernels.

V. MiAlbum Image Retrieval System

We have integrated this learning framework into the MiAlbum [12] image retrieval system developed at Microsoft Research Asia. Fig. 3 shows the user interface of this system. In this paper, we focus on image retrieval based on "query by example" and on using the user's relevance feedback and interaction to improve the system's short- and long-term performance. Fig. 4 shows the flowchart of our system. When the user submits an example image as a query, the system first computes low-level features of the query image, which are used to rank the images in the database, some of which are then shown to the user. Note that no semantic features are involved at this stage. Then, the user provides his feedback by clicking on the "thumb up" or "thumb down" button according to his judgment of the relevance of each retrieved image. With the user's relevance feedback, the system starts to take advantage of the hidden semantic features, and trains the on-line classifier to improve search performance. The search results continue to be refined iteratively until the user is satisfied. The accumulated relevance feedbacks are used to update the semantic space, as described in the long-term learning process.

TABLE I
IMAGE FEATURES USED IN OUR SYSTEM

Color-1	Color histogram in HSV space with quantization 256
Color-2	First and second moments in Lab space
Color-3	Color coherence vector in LUV space with quantization 64
Texture-1	Tamura coarseness histogram
Texture-2	Tamura directionary
Texture-3	Pyramid wavelet texture feature

VI. EXPERIMENTAL RESULTS

We performed several experiments to evaluate the effectiveness of the proposed approach on a large image database. The image database we used consists of 10 000 images of 79 semantic categories, from the Corel dataset. It is a large and heterogeneous image set. A retrieved image is considered correct if it belongs to the same category as the query image. Three types of color features and three types of texture features are used in our system; they are listed in Table I. We designed an automatic feedback scheme to model the short-term retrieval process. At each iteration, the system marks the first three incorrect images from the top 100 matches as irrelevant examples, and also selects, at most, three correct images as relevant examples (relevant examples from the previous iterations are excluded from the selection). These automatically generated feedbacks are used as training data to perform short-term learning. To model the long-term learning, we randomly select images from each category as the queries. For each query, a short-term learning process is performed and the positive feedbacks are used to construct the semantic space. That is, for each single session of retrieval, a hidden semantic feature is learned and appended as a new column to the semantic matrix. To evaluate the performance of our algorithms, we define the retrieval accuracy as follows:

$$\text{Accuracy} = \frac{\text{relevant images retrieved in top N returns}}{N}.$$

Five experiments were designed to evaluate our proposed algorithms. The experiments with the SVM training algorithm are discussed in Section VI-A. In Section VI-B, we show how the image retrieval performance improves as the semantic space is refined based on the user's interaction with the system. We further test the system's performance on the semantic space whose dimension has been reduced using SVD in Section VI-C. The system's robustness to noise is evaluated in Section VI-D.

A. SVM Learning Algorithm

We compared the performance of the SVM training algorithm with RBF kernel to the relevance feedback approach described in Rui [16]. The comparison was made in the low-level feature space, with no semantic features involved. Fig. 5 shows the fraction of relevant images among the top $N = 20$ images returned by each method, as a function of the number of rounds of user feedback. We obtained similar results using other values of N, up to 100. As can be seen, the SVM training algorithm outperformed Rui's approach in these tests.

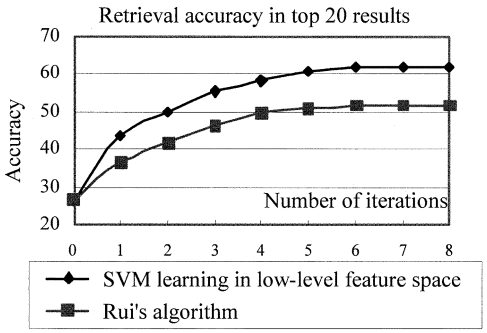

Fig. 5. Comparison of SVM learning algorithm with Rui's approach. The SVM learning algorithm outperforms Rui's approach.

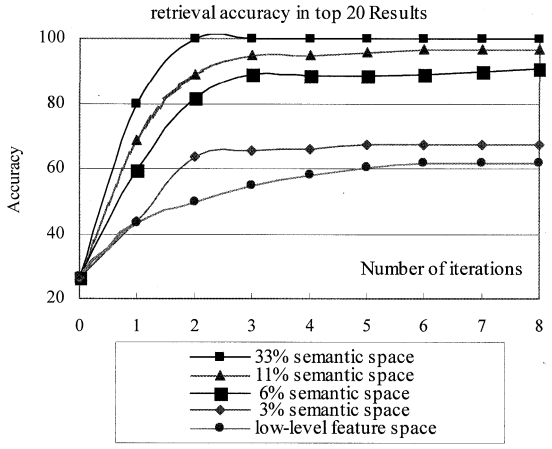

Fig. 6. Retrieval accuracy of the system improves as the semantic degree of the system increases. The graph also show the system can quickly reach a reasonably good performance with two to three iterations.

B. Image Retrieval in Boolean Semantic Space—System Evolution Evaluation

As discussed previously, the high-level semantic space is constructed as the system evolves. To evaluate the degree of system evolution, a measurement called semantic degree is defined as follows:

$$\text{Semantic degree} = \frac{\text{number of queries in semantic category}}{\text{number of images in semantic category}}.$$

For simplicity, a semantic space with semantic degree $\alpha\%$ is referred to as a $\alpha\%$ *semantic space* in this section. The semantic degree of a semantic space can be measured by the number of columns of the semantic matrix, namely, the number of hidden semantic features. For example, there are 10 000 images in our database, so a 3% semantic space corresponds to a $10\,000 \times 300$ semantic matrix before dimension reduction.

In the following, we evaluate how the system retrieval performance improves as the semantic space is learned from the user-and-system interactions. The experiments were conducted using the winnow-like mistake-driven on-line learning algorithm in Boolean semantic spaces of different semantic degrees. In Fig. 6, each curve shows the average retrieval performance. To train the system, $\alpha\%$ of the images in each category were randomly selected as query images to build the $\alpha\%$ semantic space. Then the rest of images were used as test data to evaluate the retrieval accuracy of our system at different

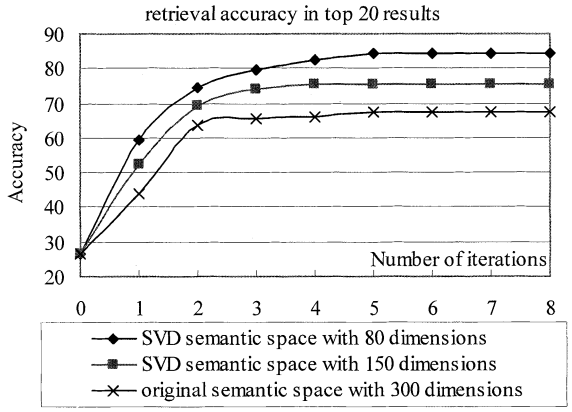

Fig. 7. Image retrieval performance in original semantic space, and dimensionality-reduced semantic space. The semantic degree of the semantic space is 3%.

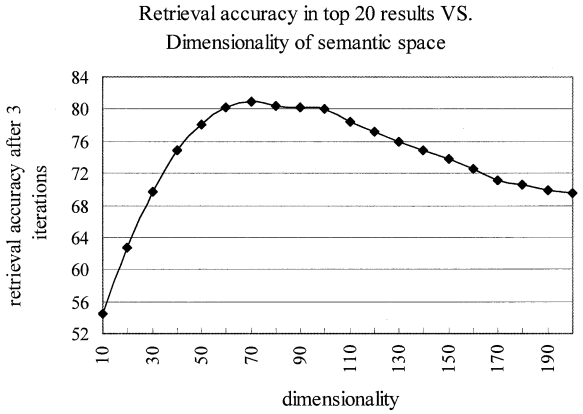

Fig. 8. Retrieval accuracy in the 3% semantic space with different degrees of dimensionality reduction. The evaluation is conducted after three iterations (the system starts to converge at this point). As can be seen, the system reaches the best performance when the number of dimensions approximates the number of semantic classes, i.e., 79.

degrees of evolution. As can be seen, the system performance improved as the semantic degree of the system increased. In addition, we see that our system learned to retrieve the target images quite quickly. It reached a reasonably good performance within two to three iterations.

C. Image Retrieval in Dimensionality-Reduced Semantic Space

In this section, we evaluate the image retrieval performance in a dimensionality-reduced semantic space. The SVD was used to reduce the dimension of the original semantic space, and then an SVM was trained to classify images in this space. After the long-term learning, a 3% semantic space represented by a 10 000 $\times$ 300 semantic matrix was constructed. Fig. 7 shows the experimental results, comparing the image retrieval performance in the original semantic space with the dimensionality-reduced spaces.

As we discussed in Section III-C, the fundamental problem for updating the semantic space and reducing its dimension is to estimate the true rank of the semantic space. The optimal rank is closely related to the number of semantic classes in the database. If the image database administrator has prior knowledge about this number, it can be used as a guideline to control the dimensionality reduction. Intuitively, the system reaches the best performance (in terms of accuracy and efficiency) when the rank of the semantic space is close to the number of semantic classes. Reducing the dimension of the semantic space to below this rank will start to cause information loss and decrease the retrieval accuracy. This intuition is supported by our experiments. As we can see from Fig. 8, the system achieved its best performance when the number of dimensions of the inferred semantic space approximated the number of semantic classes (in this case, 79).

D. Learning Semantic Space Under a Noisy Environment

In the previous experiments, the simulated user's relevance feedback was generated based on the ground truth, i.e., the 79 image categories from the Corel image library. In this case, the user is regarded as an optimal teacher. That is, the images that the user marked as positive always belonged to the same semantic class. However, in the real world, the user may make

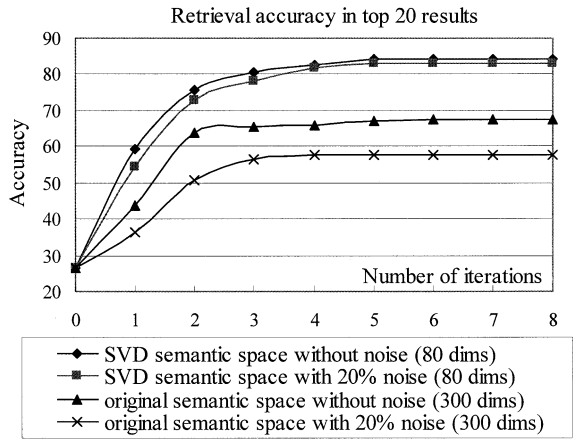

Fig. 9. Retrieval accuracy in the 3% semantic space with 20% noise. In other words, 20% of the user's feedback is incorrect. As can be seen, although the system with noise performs a little worse than the system without noise, the difference is not significant. When the rank of the semantic matrix is reduced using SVD, the performance difference becomes smaller.

mistakes in providing feedback. For instance, a user may unconsciously select images of "wolf" as positive examples while he is actually looking for images of "dog." Hence, noise could be introduced into the system when the semantic space is being constructed. The noise has two effects on the system: 1) for long-term learning, the noise will degrade the reliability of the inferred semantic space and 2) for short-term learning, the noise will mislead the current retrieval session.

In this section, we examine how the noise affects the long-term learning. We conducted experiments in which the original semantic space contained 20% noise. In other words, 20% of the simulated user's feedback was incorrect. As can be seen from Fig. 9, although the system with noise performed a little worse than the system without noise, the difference was not significant. When the rank of the semantic matrix was reduced using SVD, the performance difference became smaller. This suggests that the SVD not only reduces the dimensionality, but also helps to remove the noise introduced in the long-term learning process. After four iterations, the performance difference is less than 3%. These experiments

indicate that our proposed learning algorithms for inferring the semantic space are robust in a noisy environment, which is crucial for practical use in the real world.

VII. Conclusions and Future Work

In this paper, we described a learning framework that makes use of relevance feedback to enhance the performance of an image retrieval system from both short- and long-term perspectives. The proposed long-term learning scheme infers a semantic space from user interactions. A method of updating of the semantic space and guidelines for choosing the optimal dimensionality (rank) were also discussed. As can be seen from the experiments, this learned semantic space supplements the low-level features in making the image search result more satisfactory to users.

For the short-term learning, a winnow-like mistake-driven learning algorithm and a SVM training algorithm were used to learn the target function for retrieving relevant images from the database. A theoretical analysis of the winnow-like algorithm shows that the *mistake bound* in short-term learning is logarithmic with the total number of features, and linear with the number of relevant features. In fact, without considering the effect of the low-level features, the *semantic degree* of the constructed semantic space determines at most how many relevant images can be retrieved for a given query, and the *mistake bound* provides estimation about at most how many feedbacks are needed to retrieve *all* these relevant images. We also stated conditions under which the dimensionality-reduced semantic space is linearly separable. Based on this analysis, a SVM training algorithm was used to retrieve the target images from the database.

In our proposed learning approaches, the positive examples from the user's relevance feedback are mainly used for inferring the semantic space in the long-term learning. A possible extension of our work is to consider assigning a negative value for those negative examples while appending a new column to the semantic matrix. We are currently exploring the impact of this extension. On the other hand, as many other researchers have suggested, the negative examples—which correspond to the failure of current classifier (target function) in the short-term learning—contain the most valuable information for improving the performance in the current query session. Though our system does not explicitly direct users to provide this kind of feedback, we believe that the system will converge to a satisfactory result in fewer steps if such guidance is provided to users. Furthermore, the feedback provided by real-world users often contains inaccurate information. Although our proposed learning approaches can tolerate noise to some extent, it may be desirable to conduct filtering to remove unreliable feedback before using it for training the system.

Appendix

Proof of Theorem 1: For the sake of simplicity, we define the *relevant weights* to be $w_{i_1}, w_{i_2}, \ldots, w_{i_k}$, where the i_j are the subscripts of the corresponding relevant features. We define the total weight for each trial (each time a mistake occurs) to be $\pi^{(t)} = w_1^{(t)} + w_2^{(t)} + \cdots + w_n^{(t)}$, where $\boldsymbol{w}^{(t)}$ is the weight vector in the tth trial. Any positive mistake will increase at least one relevant weight. And a negative mistake will not decrease any of the relevant weights. Furthermore, each of these relevant weights can be increased at most $1 + \log_\alpha \theta$ times. Therefore, the algorithm makes at most $M_p = k(1 + \log_\alpha \theta)$ positive mistakes. For each positive mistake, the weight $w_i^{(t)}$ increases if the corresponding $x_i^{(t)}$ equals one. Therefore, the total weight $\pi^{(t)}$ increases by at most $Y = n + (\alpha - 1)\theta$.

On the other hand, each negative mistake decreases the total weight by at least $Z = ((\alpha - 1)/\alpha)\theta$. Let M_N denote the number of negative mistakes. Thus, $n + M_P Y - M_N Z \geq 0$. This leads to the upper bound on the number of negative mistakes $M_N = 1/Z(n + M_P Y)$. Therefore, the total number of mistakes is bounded by

$$\begin{aligned} E &= M_P + M_N \\ &= \frac{\alpha n}{(\alpha - 1)\theta} + \left[\frac{\alpha n}{(\alpha - 1)\theta} + \alpha + 1\right] k(1 + \log_\alpha \theta). \end{aligned}$$

Proof of Theorem 2: ($\Rightarrow$) If $S_{\mathbf{w}} \cap S_V \neq \phi$, then $\exists \mathbf{w}$, $\mathbf{w} \in S_{\mathbf{w}}$ and $\mathbf{w} \in S_V$, such that $\mathbf{w}$ is a separator in the original semantic space. The hyperplane discriminating the relevant images and irrelevant images in the original semantic space is

$$\mathbf{w}\mathbf{x}^T = 0.$$

Note that $\mathbf{w}$ and $\mathbf{x}$ are both row vectors. Since $\mathbf{w} \in S_V$, we have $\mathbf{w} = \mathbf{w}'V^T$. Thus

$$\begin{aligned} \mathbf{w}\mathbf{x}^T = 0 &\Rightarrow \mathbf{w}'V^T\mathbf{x}^T = \mathbf{w}'(\mathbf{x}V)^T = 0 \\ &\Rightarrow \mathbf{w}'\mathbf{y}^T = 0 \end{aligned}$$

where $\mathbf{y} = \mathbf{x}V$ is the image vector in the reduced semantic space. Thus, the relevant and irrelevant images are still linearly separable in the reduced semantic space.

($\Leftarrow$) If the relevant images and irrelevant images are still linearly separable in the reduced semantic space, we assume that $\mathbf{w}'$ is such a separator. Thus

$$\mathbf{w}'\mathbf{y}^T = 0$$

is a discriminating hyperplane in the reduced semantic space. Since $\mathbf{y} = \mathbf{x}V$, we have

$$\mathbf{w}'(\mathbf{x}V)^T = 0 \Rightarrow (\mathbf{w}'V^T)\mathbf{x}^T = 0.$$

Thus, $\mathbf{w} = \mathbf{w}'V^T$ is a linear separator in the original semantic space. That is, $\mathbf{w} \in S_{\mathbf{w}}$. Obviously, $\mathbf{w} \in S_V$. Therefore, $S_{\mathbf{w}} \cap S_V \neq \phi$.

References

[1] M. W. Berry, S. M. Dumais, and G. W. O'Brien, "Using linear algebra for intelligent information retrieval," *SIAM Rev.*, vol. 37, no. 4, pp. 573–595, 1995.

[2] Z. Chen, X. Meng, B. Zhu, and R. Fowler, "Websail: From on-line learning to web-search," in *Proc. 1st Int. Conf. Web Information Systems Engineering*, vol. 1, Hong Kong, China, June 2000, pp. 192–199.

[3] I. J. Cox, J. Ghosn, M. L. Miller, T. V. Papathomas, and P. N. Yianilos, "Hidden annotation in content based image retrieval," in *Proc. IEEE Workshop Content-Based Access of Image and Video Libraries*, 1997, pp. 76–81.

[4] I. J. Cox, T. P. Minka, T. V. Papathomas, and P. N. Yianilos, "The Bayesian image retrieval system, PicHunter: Theory, implementation, and psychophysical experiments," *IEEE Trans. Image Processing*, vol. 9, pp. 20–37, Jan. 2000.

[5] I. Dagan, Y. Karov, and D. Roth, "Mistaken-driven learning in text categorization," in *Proc. 2nd Conf. Empirical Methods in Natural Language Processing*, 1997, pp. 55–63.
[6] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The QBIC system," *IEEE Computer*, vol. 28, pp. 23–32, Sept. 1995.
[7] Y. Ishikawa, R. Subramanya, and C. Faloutsos, "MindReader: Query databases through multiple examples," in *Proc. 24th Int. Conf. Very Large DataBases*, New York, 1998, pp. 218–227.
[8] J. Kivinen, M. K. Warmuth, and P. Auer, "The perceptron algorithm vs. winnow: Linear vs. logarithmic mistake bounds when few input variables are relevant," *Artific. Intell.*, pp. 325–343, 1997.
[9] MAC. Lee, W. Y. Ma, and H. J. Zhang, "Information embedding based on user's relevance feedback for image retrieval," in *Multimedia Storage and Archiving Systems IV*, Boston, Sept. 1999.
[10] D. Lewis, R. E. Schapire, J. P. Callan, and R. Papka, "Training algorithms for linear text classifier," in *Proc. 19th Annu. Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 1996, pp. 298–306.
[11] N. Littlestone, "Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm," *Mach. Learn.*, vol. 2, no. 4, pp. 285–318, 1988.
[12] W. Liu, Y. Sun, and H. J. Zhang, "MiAlbum—A system for home photo management using the semi-automatic image annotation approach," in *Proc. ACM Multimedia 2000*, Los Angeles, CA, Oct.–Nov. 2000.
[13] W. Y. Ma and B. S. Manjunath, "Netra: A toolbox for navigating large image databases," *ACM Multimedia Syst.*, vol. 7, pp. 184–198, 1999.
[14] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala, "Latent semantic indexing: A probabilistic analysis," in *Proc. 17th ACM Symp. Principle of Database Systems*, Seattle, WA, 1998, pp. 159–168.
[15] H. Ragas and C. H. Koster, "Four text classification algorithms compared on a Dutch corpus," in *Proc. SIGIR-98, 21st ACM Int. Conf. Research and Development in Information Retrieval*, Melbourne, Australia, 1998, pp. 369–370.
[16] Y. Rui and T. S. Huang, "A novel relevance feedback techniques in image retrieval," in *ACM Multimedia*, 1999, pp. 67–70.
[17] Y. Rui, T. S. Huang, S. Mehrotra, and M. Ortega, "A relevance feedback architecture for content-based multimedia information retrieval systems," in *Proc. IEEE Workshop Content-based Access of Image and Video Libraries*, 1997, pp. 82–89.
[18] G. Salton and M. McGill, *Introduction to Modern Information Retrieval*. New York: McGraw-Hill, 1983.
[19] S. Santini and R. Jain, "Visual navigation in perceptual databases," in *Proc. 1997 Int. Conf. Visual Information Systems*, San Diego, CA, Dec. 1997.
[20] K. Tieu and P. Viola, "Boosting image retrieval," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Hilton Head Island, SC, June 2000, pp. 228–235.
[21] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proc. ACM Multimedia 2001*, Ottawa, ON, Canada, Sept. 2001.
[22] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer, 1995.
[23] Y. Wu, Q. Tian, and T. S. Huang, "Discriminant-EM algorithm with application to image retrieval," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Hilton Head Island, SC, June 2000.
[24] X. S. Zhou and T. S. Huang, "Comparing discriminating transformations and SVM for learning during multimedia retrieval," in *Proc. ACM Multimedia 2001*, Ottawa, ON, Canada, 2001.
[25] ——, "BiasMap for small sample learning during multimedia retrieval," in *Proc. IEEE Computer Vision and Pattern Recognition*, Kauai, HI, Dec. 2001.

Xiaofei He received the B.S. degree in computer science from Zhejiang University, Zhejiang, China, in 2000. He is currently working toward the Ph.D. degree in the Department of Computer Science, University of Chicago, Chicago, IL.

His research interests are machine learning, pattern recognition, dimensionality reduction, and image retrieval.

Oliver King received the M.S. degree in computer science in 1999 and the Ph.D. degree in mathematics in 2001, both from the University of California at Berkeley.

He is currently a Postdoctoral Fellow at Harvard Medical School, Cambridge, MA.

Wei-Ying Ma received the B.S. degree in electrical engineering from the National Tsing Hua University, Taiwan, R.O.C., in 1990, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at Santa Barbara in 1994 and 1997, respectively.

His research interests include content-based image and video retrieval, machine learning, information retrieval, and adaptive content delivery. He serves as an Associate Editor for the *Journal of Multimedia Tools and Applications*, has published four book chapters, and has served on the organizing and program committees of several international conferences.

Mingjing Li received the B.S. degree in electrical engineering from the University of Science and Technology of China, Hong Kong, in 1989 and the Ph.D. degree in pattern recognition from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1995.

He joined Microsoft Research Asia, Beijing, China, in July 1999. His research interests include handwriting recognition, statistical language modeling, search engineers, and multimedia information retrieval.

Hong-Jiang Zhang (M'91–SM'97) received the B.S. degree from Zhengzhou University, Zhengzhou, China, in 1982 and the Ph.D. degree from the Technical University of Denmark, Lyngby, in 1991, both in electrical engineering.

His research interests include video and image analysis and processing, content-based image/video/audio retrieval, media compression and streaming, computer vision, and their applications in consumer and enterprise markets. He has published over 120 papers in these areas. He serves on the edtiorial boards of five professional journals and a dozen committees for various international conferences.