Asymmetric Bagging and Random Subspace for Support Vector Machines-Based Relevance Feedback in Image Retrieval

Dacheng Tao, *Student Member*, *IEEE*, Xiaoou Tang, *Senior Member*, *IEEE*, Xuelong Li, *Member*, *IEEE*, and Xindong Wu, *Senior Member*, *IEEE*

Abstract—Relevance feedback schemes based on support vector machines (SVM) have been widely used in content-based image retrieval (CBIR). However, the performance of SVM-based relevance feedback is often poor when the number of labeled positive feedback samples is small. This is mainly due to three reasons: 1) an SVM classifier is unstable on a small-sized training set, 2) SVM's optimal hyperplane may be biased when the positive feedback samples are much less than the negative feedback samples, and 3) overfitting happens because the number of feature dimensions is much higher than the size of the training set. In this paper, we develop a mechanism to overcome these problems. To address the first two problems, we propose an asymmetric bagging-based SVM (AB-SVM). For the third problem, we combine the random subspace method and SVM for relevance feedback, which is named random subspace SVM (RS-SVM). Finally, by integrating AB-SVM and RS-SVM, an *asymmetric bagging and random subspace* SVM (ABRS-SVM) is built to solve these three problems and further improve the relevance feedback performance.

Index Terms—Classifier committee learning, content-based image retrieval, relevance feedback, asymmetric bagging, random subspace, support vector machines.

1 Introduction

Relevance feedback [21] is an important tool to improve the performance of content-based image retrieval (CBIR) [22]. In a relevance feedback process, the user first labels a number of relevant retrieval results as positive feedback samples and some irrelevant retrieval results as negative feedback samples. Then, a CBIR system refines all retrieval results based on these feedback samples. These two steps are carried out iteratively to improve the performance of the image retrieval system by gradually learning the user's preferences.

Many relevance feedback methods have been developed in recent years. They either adjust the weights of various features to adapt to the user's preferences [21] or estimate the density of the positive feedback examples [5]. Moreover, discriminant learning has also been used as a feature selection method for relevance feedback [32]. These methods work well with certain limitations. The method in [1] is heuristic. The density estimation method in [5] loses information contained in negative samples. The discriminant learning in [32] often suffers from the matrix singular problem.

- *D. Tao and X. Li are with the School of Computer Science and Information Systems, Birkbeck, University of London, Malet Street, London WC1E 7HX, UK. E-mail: {dacheng, xuelong}@dcs.bbk.ac.uk.*
- *X. Tang is with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong. E-mail: xtang@ie.cuhk.edu.hk.*
- *X. Wu is with the Department of Computer Science, University of Vermont, 33 Colchester Avenue, Burlington, VT 05405. E-mail: xwu@cs.uvm.edu.*

Manuscript received 4 Jan. 2005; revised 17 Aug. 2005; accepted 30 Sept. 2005; published online 11 May 2006.
Recommended for acceptance by A. Rangarajan.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0010-0105.

Regarding the positive samples and the negative samples as two difference groups and aiming at finding a classifier to identify these two groups from each other, relevance feedback in CBIR becomes an online learning problem. In other words, it is a real-time classification problem. Recently, classification-based relevance feedback [8], [31], [27], [6], [25] has become a popular topic in CBIR. Many binary-classification algorithms were designed to treat the positive samples and negative samples equally [31], [27]. In One-Class SVM [5], the algorithm only estimates the data density of the positive samples. Zhou and Huang [32] also proposed an (1 + x)-class classification in which there is an unknown number of classes but the user only concerns one class (positive feedback samples). Multiclass-based relevance feedback [18] was also developed to scale the performance of this key part of a CBIR system.

Among these classifiers, the Support Vector Machines (SVM) [8], [31], [27], [6], [25] based relevance feedback (SVM RF) has shown promising results owing to its good generalization ability. SVM has a very good performance for pattern classification problems by minimizing the Vapnik-Chervonenkis dimensions and achieving a minimal structural risk [29]. SVM active learning [27] halves the image space each time in which the most positive samples are selected farthest from the classifier boundary on the positive side and the samples close to the boundary are deemed as the most-informative ones for the user to label. Guo et al. [6] developed a constrained similarity measure (CSM) for image retrieval in which the SVM is also employed with AdaBoost. The CSM also learns a boundary that halves the images in the database into two groups and images inside the boundary are ranked by their Euclidean distances to the query. There are also some more kinds of SVM-based relevance feedback algorithms [33]. However, when the number of positive feedback

samples is small, the performance of SVM RF becomes poor. This is mainly due to the following three reasons:

- First, an SVM classifier is unstable for a small-sized training set, i.e., the optimal hyperplane of SVM is sensitive to the training samples when the size of the training set is small. In SVM RF, the optimal hyperplane is determined by the feedback samples. However, more often than not, the users would only label a few images and cannot label each feedback sample accurately all the time. Therefore, the performance of the system may be poor with insufficient and inexactly labeled samples.
- Second, SVM's optimal hyperplane may be biased when the positive feedback samples are much less than the negative feedback samples. In the relevance feedback process, there are usually many more negative feedback samples than positive ones. Because of the imbalance of the training samples for the two classes, SVM's optimal hyperplane will be biased toward the negative feedback samples. Consequently, SVM RF may mistake many query-irrelevant images as relevant ones.
- Finally, in relevance feedback, the size of the training set is much smaller than the number of dimensions in the feature vector and, thus, may cause an overfitting problem. Because of the existence of noise, some features can only discriminate the positive and negative feedback samples but cannot distinguish the relevant or irrelevant images in the database. So, the learned SVM classifier, which is based on the feedback samples, cannot work well for the remaining images in the database.

In order to overcome these problems, we design a set of new algorithms to improve the SVM RF for CBIR. The key idea comes from the classifier committee learning (CCL) [2], [7], [12]. Since each classifier has its own unique ability and property to classify relevant and irrelevant samples, the CCL can pool a number of weak classifiers to improve the recognition performance. We use bagging and a random subspace method to improve the SVM since they are especially effective when the original classifier is not very stable.

The rest of the paper is organized as follows: In Section 2, SVM-based relevance feedback is briefly introduced. In Section 3, we propose three algorithms based on SVM and CCL, and in Section 4, we present our image retrieval system. A large number of experiments are given in Sections 5 (with toy problems) and Section 6 (using the Corel Photo Gallery with 17,800 images). Related work is discussed in Section 7 and the conclusion is drawn in Section 8.

2 Background: SVM-Based Relevance Feedback in Content-Based Image Retrieval

SVM [29], [3] is a very effective binary classification algorithm. Consider a linearly separable binary classification problem (as shown in Fig. 1):

$$\{(x_i, y_i)\}_{i=1}^{N} \text{ and } y_i = \{+1, -1\}, \tag{1}$$

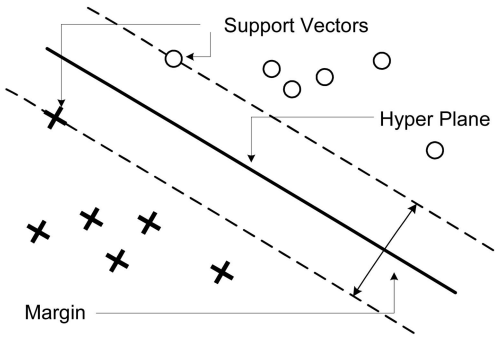


Fig. 1. SVM for a linearly separable binary classification problem.

where x_i is an n-dimension vector and y_i is the label of the class that the vector belongs to. SVM separates the two classes of points by a hyperplane,

$$w^T x + b = 0, \tag{2}$$

where $\mathbf{w}$ is an input vector, $\mathbf{x}$ is an adaptive weight vector, and b is a bias. SVM finds the parameters $\mathbf{w}$ and b for the optimal hyperplane to maximize the geometric margin $2/\parallel \mathbf{w} \parallel$, subject to

$$y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) \geq +1. \tag{3}$$

The solution can be found through a Wolfe dual problem with the Lagrangian multiplied by α_i:

$$Q(\alpha) = \sum_{i=1}^{m} \alpha_i - \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)/2, \tag{4}$$

subject to $\alpha_i \geq 0$ and $\sum_{i=1}^{m} \alpha_i y_i = 0$.

In the dual format, data points only appear in the inner product. To get a potentially better representation of the data, the data points are mapped into a Hilbert Inner Product space through a replacement:

$$\mathbf{x}_i \cdot \mathbf{x}_j \rightarrow \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j), \tag{5}$$

where $K(.)$ is a kernel function. We then get the kernel version of the Wolfe dual problem:

$$Q(\alpha) = \sum_{i=1}^{m} \alpha_i - \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)/2. \tag{6}$$

Thus, for a given kernel function, the SVM classifier is given by

$$F(\mathbf{x}) = \operatorname{sgn}(f(\mathbf{x})), \tag{7}$$

where $f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$ is the output hyperplane decision function of the SVM.

In general, when $|f(\mathbf{x})|$ for a given pattern is high, the corresponding prediction confidence will be high. Meanwhile, a low $|f(\mathbf{x})|$ of a given pattern means that the pattern is close to the decision boundary and its corresponding prediction confidence will be low. Consequently, the output of SVM, $f(\mathbf{x})$, has been used to measure the dissimilarity [8], [31] between a given pattern and the query image, in traditional SVM-based CBIR RF.

TABLE 1
Algorithm of Asymmetric Bagging SVM (AB-SVM)

Input: positive training set $\mathbf{S}^+$, negative training set $\mathbf{S}^-$, weak classifier I (SVM), integer T (number of generated classifiers), and the test sample $\mathbf{x}$.

1. For $i = 1$ to T {
2. $\quad \mathbf{S}_i^-$ = bootstrap sample from $\mathbf{S}^-$, with $|\mathbf{S}_i^-| = |\mathbf{S}^+|$.
3. $\quad C_i = I(\mathbf{S}_i^-, \mathbf{S}^+)$
4. }
5. $C^*(\mathbf{x}) = aggregation\{C_i(\mathbf{x}, \mathbf{S}_i^-, \mathbf{S}^+), 1 \le i \le T\}$

Output: classifier C^*.

3 Three Classifier Committee Learning (CCL) Algorithms for SVMs

To address the three problems of SVM-based relevance feedback described in the introduction, we propose three algorithms in this section.

3.1 Asymmetric Bagging SVM (AB-SVM)

Bagging [2] incorporates the benefits of bootstrapping and aggregation. Multiple classifiers can be generated by training on multiple sets of samples that are produced by bootstrapping, i.e., random sampling with replacement on the training samples. Aggregation of the generated classifiers can then be implemented by majority voting [1].

Experimental and theoretical results have shown that bagging can improve a *good but unstable* classifier significantly [2]. This is exactly the case of the first problem of SVM RF. However, directly using bagging in SVM RF is not appropriate since we have only a very small number of positive feedback samples. To overcome this problem, we develop a novel asymmetric bagging strategy. The bootstrapping is executed only on the negative feedback samples since there are far more negative feedback samples than the positive feedback samples. This way each generated classifier will be trained on a balanced number of positive and negative samples, thus solving the second problem as well. The asymmetric bagging SVM (AB-SVM) algorithm is described in Table 1.

In AB-SVM, the aggregation is implemented by the Majority Voting Rule (MVR). The asymmetric bagging strategy solves the unstable problem of SVM classifiers and the unbalance problem in the training set. However, it cannot solve the small sample-size problem. We will solve it by the Random Subspace Method (RSM) in the next section.

3.2 Random Subspace SVM (RS-SVM)

Similar to bagging, the random subspace method (RSM) [7] also benefits from bootstrapping and aggregation. However, unlike bagging that bootstraps training samples, RSM performs the bootstrapping in the feature space.

For SVM RF, overfitting happens when the training set is relatively small compared to the high dimensionality of the feature vector. In order to avoid overfitting, we sample a small subset of features to reduce the discrepancy between the training data size and the feature vector length. Using a random sampling method, we construct a multiple number of SVMs. We then combine these SVMs to construct a more powerful classifier to solve the overfitting problem. Our RSM-based random subspace SVM (RS-SVM) is described in Table 2.

TABLE 2
Algorithm of Random Subspace SVM (RS-SVM)

Input: feature set $\mathbf{F}$, weak classifier I (SVM), integer T (number of generated classifiers), and the test sample $\mathbf{x}$.

1. For $i = 1$ to T {
2. $\quad \mathbf{F}_i$ = bootstrap feature from $\mathbf{F}$.
3. $\quad C_i = I(\mathbf{F}_i)$
4. }
5. $C^*(\mathbf{x}) = aggregation\{C_i(\mathbf{x}, \mathbf{F}_i), 1 \le i \le T\}$

Output: classifier C^*.

TABLE 3
Algorithm of Asymmetric Bagging and Random Subspace SVM (ABRS-SVM)

Input: positive training set $\mathbf{S}^+$, negative training set $\mathbf{S}^-$, feature set $\mathbf{F}$, weak classifier I (SVM), integer T_s (the number of bagging classifiers), integer T_f (number of RSM classifiers), and the test sample $\mathbf{x}$.

1. For $j = 1$ to T_s {
2. $\quad \mathbf{S}_j^-$ = a bootstrap sample from $\mathbf{S}^-$.
3. $\quad$ For $i = 1$ to T_f {
4. $\quad\quad \mathbf{F}_i$ = a bootstrap sample from $\mathbf{F}$.
5. $\quad\quad C_{i,j} = I(\mathbf{F}_i, \mathbf{S}_j^-, \mathbf{S}^+)$
6. $\quad$ }
7. }
8. $C^*(\mathbf{x}) = aggregation\left\{ \begin{matrix} C_{ij}(\mathbf{x}, \mathbf{F}_i, \mathbf{S}_j^-, \mathbf{S}^+) \\ 1 \le i \le T_f, 1 \le j \le T_s \end{matrix} \right\}$.

Output: classifier C^*.

3.3 Asymmetric Bagging and Random Subspace SVM (ABRS-SVM)

Since the asymmetric bagging method can overcome the first two problems of SVM RF and the RSM can overcome the third problem of the SVM RF, we should be able to integrate the two methods to solve all the three problems together. So, we propose an *asymmetric bagging and random subspace* SVM (ABRS-SVM) to combine the two. The algorithm is described in Table 3.

In order to explain why the *asymmetric bagging and random subspace* strategy works, we derive the proof following a similar discussion on bagging in [2].

Let $(y, \mathbf{x})$ be a data sample in the training set L with a feature vector F, where y is the class label of the sample $\mathbf{x}$ and L is drawn from a probability distribution P. Suppose $\varphi(\mathbf{x}, L, F)$ is a simple predictor (classifier) constructed by the *asymmetric bagging and random subspace* strategy and the aggregated predictor is

$$\varphi_A(\mathbf{x}, P) = E_F E_L \varphi(\mathbf{x}, L, F).$$

Let random variables $(Y, \mathbf{X})$ be drawn from the distribution independent of the training set L. The average predictor error, estimated by $\varphi(\mathbf{x}, L, F)$, is

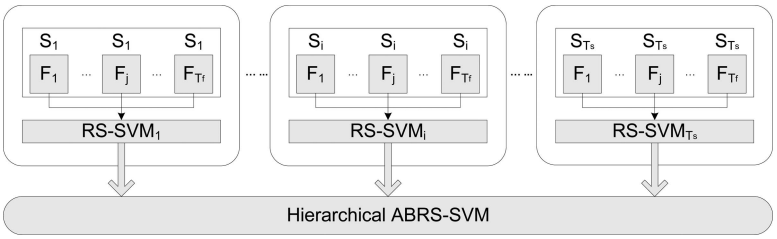


Fig. 2. Hierarchical structure of aggregation.

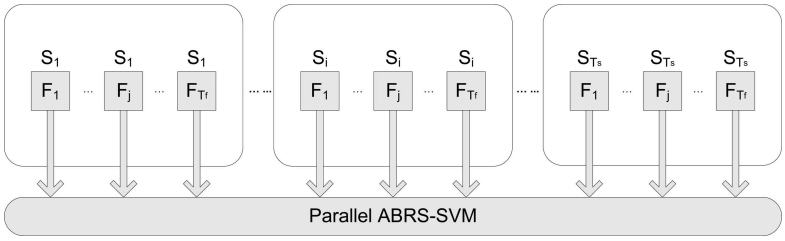


Fig. 3. Parallel structure of aggregation.

$e_a = E_F E_L E_{Y,\mathbf{X}}(Y - \varphi(\mathbf{X}, L, F))^2$. The corresponding error estimated by the aggregated predictor is

$$e_A = E_{Y,\mathbf{X}}(Y - \varphi_A(\mathbf{X}, P))^2. \tag{8}$$

Using the inequality,

$$\frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N}(z_{ij})^2 \geq \left(\frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N} z_{ij}\right)^2,$$

we have

$$E_F E_L \varphi^2(\mathbf{X}, L, F) \geq (E_F E_L \varphi(\mathbf{X}, L, F))^2. \tag{9}$$

Through (9), we have

$$E_{Y,X} E_F E_L \varphi^2(\mathbf{X}, L, F) \geq E_{Y,X}\varphi_A^2(\mathbf{X}, P). \tag{10}$$

Thus, from (8) and (10), we can obtain

$$\begin{aligned} e_a &= E_{Y,X}Y^2 - 2E_{Y,X}Y\varphi_A + E_{Y,X}E_F E_L \varphi^2(\mathbf{X}, L, F) \\ &\geq E_{Y,X}(Y - \varphi_A)^2 = e_A. \end{aligned} \tag{11}$$

Here, we have made an assumption that the average performance of all the individual classifiers $\varphi(\mathbf{x}, L, F)$, each trained on a subset of features and the training set replicas, is similar to a classifier which uses the full feature set and the whole training set.

This can be true when the sizes of features and training data are adequate to approximate the full data distribution. Even when this is not true, the drop of accuracy for each single classifier may be well compensated in the aggregation process.

As in (11), from the inequality, we can see that the more diverse is the $\varphi(\mathbf{x}, L, F)$, the more accurate is the aggregated predictor. Practically, the aggregated predictor is not $\varphi_A(\mathbf{x}, P)$, but $\varphi_A(\mathbf{x}, P')$ because the *asymmetric bagging and random subspace* strategy is used on the training set. P' and P are consistent in the probability space.

If the classifier φ is stable, $\varphi_A(\mathbf{x}, P')$ (which approximates $\varphi(\mathbf{x}, L, F)$) given by the *asymmetric bagging and random subspace* strategy is not as accurate as $\varphi(\mathbf{x}, P)$. Therefore, the strategy may not work. However, if is unstable (because weak classifiers are diverse), $\varphi(\mathbf{x}, P')$ can improve the performance.

In CBIR RF, SVM classifiers are unstable both for the training features and for the training samples. Consequently, the *asymmetric bagging and random subspace* strategy can improve the performance to generate the *asymmetric bagging and random subspace* SVM (ABRS-SVM).

There are many different ways to do the aggregation, two of which are hierarchical and parallel structures. The hierarchical ABRS-SVM (HABRS-SVM) structure of the aggregation is shown in Fig. 2 and Fig. 3 illustrates the parallel ABRS-SVM (PABRS-SVM) structure.

For a given pattern, we first recognize it by a series of weak SVMs, which are constructed by the bootstrapping training set and features and are denoted as $\{C_{ij} = C(F_i, S_j) | 1 \leq i \leq T_f, 1 \leq j \leq T_s\}$. Then, we recognize it on a subset of weak classifiers $\{C_i = C(F_i, S_j) | 1 \leq i \leq T_f\}$, which are constructed on the same training examples, but with different training features. At last, we use these outputs and the aggregation rule to construct the destination classifier. For example, if the aggregation rule is majority voting and the weak classifier $C_{ij}(\mathbf{x}, \mathbf{F}i, \mathbf{S}j) \in \{0, 1\}$, we can represent it as:

$$C^*(\mathbf{x}) = \operatorname{sgn}\left\{\sum_j \operatorname{sgn}\left[\sum_i C_{ij}(\mathbf{x}, \mathbf{F}i, \mathbf{S}j) - \frac{T_f - 1}{2}\right] - \frac{T_s - 1}{2}\right\}. \tag{12}$$

The parallel structure of the aggregation is shown in Fig. 3.

For a given pattern, we recognize it by all weak SVMs $\{C_{ij} = C(F_i, S_j) | 1 \leq i \leq T_f, 1 \leq j \leq T_s\}$. Then, an aggregation rule is utilized to classify it as query relevant or irrelevant. For example, if the aggregation rule is majority voting and the weak classifier $C_{ij}(\mathbf{x}, \mathbf{F}i, \mathbf{S}j) \in \{0, 1\}$, we can represent it as:

$$C^*(\mathbf{x}) = \operatorname{sgn}\left\{\sum_{i,j} C_{ij}(\mathbf{x}, \mathbf{F}i, \mathbf{S}j) - \frac{T_s T_f - 1}{2}\right\}. \tag{13}$$

Since the *asymmetric bagging and random subspace* strategy can generate more diversified classifiers than using bagging or RSM alone, it should outperform the two. In order to

achieve the maximum diversity, we choose to combine all generated classifiers in parallel as shown in Fig. 3. The experiments (Figs. 10, 11, 12, and 13 in Section 6.3) show that this is better than combining bagging or RSM first and then combing RSM or bagging.

In summary, for a given test sample, we first recognize it by all $T_f \cdot T_s$ weak SVM classifiers:

$$\{C_{ij} = C(F_i, S_j) | 1 \le i \le T_f, 1 \le j \le T_s\}. \quad (14)$$

Then, an aggregation rule is used to integrate all the results from the weak classifiers for final classification of the sample as either relevant or irrelevant.

3.4 Aggregation Model

After training a given classifier committee learning (CCL) model, such as AB-SVM or RS-SVM, an aggregation rule should be given to combine the weak classifiers. Many aggregation models have been developed, such as the MVR [1], Bayes sum rule (BSR) [12], Bayes product rule [12], LSE-based weighting rule [11], double-layer combination [10], Dempster-Shafer model [28], and some nonlinear methods. In this paper, we only focus on the MVR and the BSR, due to their good performance in pattern classification.

3.4.1 Majority Voting Rule (MVR)

MVR is the simplest method to combine multiple classifiers. Given a series of weak classifiers $\{Ci(\mathbf{x}), 1 \le i \le N\}$, the MVR can be represented as follows:

$$C^*(\mathbf{x}) = \operatorname{sgn}\left\{\sum_i C_i(\mathbf{x}) - \frac{N-1}{2}\right\}. \quad (15)$$

MVR does not consider any individual behavior of each weak classifier. It only counts the largest number of classifiers that agree with each other.

3.4.2 Bayes Sum Rule (BSR)

MVR does not consider the behaviors of the weak classifiers. If one classifier is much more accurate than all the others, the MVR cannot take advantage of it. To address this problem, Kittler et al. [12] proposed a general theoretical framework based on the Bayesian decision rule. We select BSR in this paper to aggregate multiple classifiers because BSR outperform most of the other rules.

Let $z_i (1 \le i \le R)$ be the ith classifier, where R is the number of classifiers. In the BSR measurement space, each class Y_k is modeled by the probability density function (PDF) $p(z_i \mid y_k)$. Assuming its priori-probability is $p(y_k)$, BSR combines z_i as follows:

$$C^*(\mathbf{x}) = \arg\max_k \left[(1 - \mathrm{R})P(y_k) + \sum_{i=1}^{R} P(y_k|z_i)\right]. \quad (16)$$

To use the BSR in our schemes (AB-SVM, RS-SVM, and ABRS-SVM), a probability model is required. As shown in [19], the sigmoid function combined with the output of SVM can be used to estimate the class-conditional probability for a given instance $\mathbf{x}$ by

$$P(y_k|z_i) = 1/\{1 + \exp(-|f_i(\mathbf{x})|)\}. \quad (17)$$

TABLE 4
Algorithms' Computational Complexity

	Training	Testing
SVM	$O(SVM)$	$n_s^{SVM} \cdot n_f^{SVM} \cdot (\otimes + \oplus)$
AB-SVM	$T_s \cdot O(SVM)$	$T_s \cdot n_s^{AB\text{-}SVM} \cdot n_f^{AB\text{-}SVM} \cdot (\otimes + \oplus)$
RS-SVM	$T_f \cdot O(SVM)$	$T_f \cdot n_s^{RS\text{-}SVM} \cdot n_f^{RS\text{-}SVM} \cdot (\otimes + \oplus)$
ABRS-SVM	$T_s \cdot T_f \cdot O(SVM)$	$T_s \cdot T_f \cdot n_s^{ABRS\text{-}SVM} \cdot n_f^{ABRS\text{-}SVM} \cdot (\otimes + \oplus)$

We do not need to consider $p(y_k)$ here because the probability for an unknown sample to be query relevant or irrelevant is equal. Then, BSR is simplified as follows:

$$C^*(\mathbf{x}) = \arg\max_k \left[\sum_{i=1}^{R} P(y_k|z_i)\right]. \quad (18)$$

3.5 Dissimilarity Measure

3.5.1 Using MVR to Combine SVMs (MVR-SVM)

For a given sample, we first use the MVR to recognize it as query relevant or irrelevant. Then, we measure the dissimilarity between the sample and the query as the output of the individual SVM classifier, which gives the same label as the MVR and produces the highest confidence value (the absolute value of the decision function of the SVM classifier).

3.5.2 Using BSR to Combine SVMs (BSR-SVM)

For a given sample, we first use the BSR to recognize it as query relevant or irrelevant. Then, we measure the dissimilarity between the sample and the query using the individual SVM classifier, which gives the same label as the BSR and has the highest confidence value.

3.5.3 Bayes Sum Rule (BSR)

From the definition of the BSR, the output of the BSR $\sum_{i=1}^{R} P(y_k|x_i)$ can also be used as a dissimilarity measure between a given sample and the query.

In this paper, we will compare all the three rules for ABRS-SVM-based relevance feedback.

3.6 Computational Complexity Analysis

From [3], we know that the computational complexity for training a SVM is $O(SVM) = O(n_s^3 + n_s^2 L + n_s n_f L)$, where n_s is the number of support vectors, n_f is the number of feature dimensions, and L is the size of the training set. From the formula of the output of SVM, the number of the support vectors n_s determines the computational complexity in the testing stage. We denote the computational complexity for a multiplication and addition of two real values as $\bigotimes$ and $\bigoplus$, respectively. Then the computational complexities of SVM, AB-SVM, RS-MSVM, and ABRS-SVM are given in Table 4.

4 The Content-Based Image Retrieval System

CBIR assumes that the user expects the best possible retrieval results after each relevance feedback iteration, i.e., the search engine is required to return the most semantically relevant images based on the previous feedback samples. At the same time, the user is impatient and 1) will never label a significant number of images for each relevance feedback iteration and

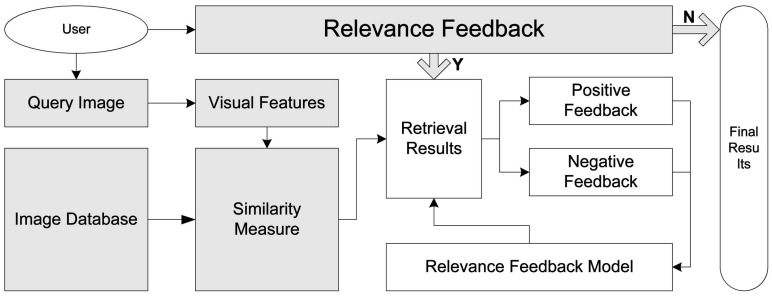


Fig. 4. Image retrieval system flow chart.

2) only does a few numbers of iterations. To deal with this type of scenario, the following CBIR framework is proposed. With the proposed system, we can embed many kinds of relevance feedback algorithms easily.

From Fig. 4, when a query (image) is given, the low-level visual features are extracted. Then, all images in the database are sorted based on a similarity metric. If the user is satisfied with the results, the retrieval process is ended. However, most of the time, the relevance feedback is needed because of a poor retrieval performance. The user labels some top images as positive feedback samples or negative feedback samples. Using these feedback samples, a relevance feedback model can be trained based on certain machine learning algorithms. The similarity metric can be updated as well as the relevance feedback model. Then, all the images are resorted based on the recalculated similarity metric. If the user is not satisfied with the results, the relevance feedback process will be performed iteratively.

This system implements the ABRS-SVM algorithm in Section 3 and compares its accuracy with several *state-of-the-art* relevance feedback algorithms. In the system, images are represented by three main features: color [23], [17], [14], [9], [16], texture [14], [24], [13], [15], [4], [30], and shape [9], [16]. The color information is the most informative feature for image retrieval because of its robustness with respect to scaling, rotation, perspective, and occlusion of the image [23]. The texture information is another important cue for image retrieval. Previous studies have shown that texture information according to the structure and orientation fits well with the model of human perception and so does the shape information. Details of the three features are given as follows:

- **Color**. A 256-bin HSV color histogram [23] is selected. Hue and saturation are both quantized into eight bins, while value is quantized into four bins.
- **Texture**. The system selects the pyramidal wavelet transform (PWT) from the Y component in the YCrCb space for texture representation. An image is decomposed by the traditional PWT with Haar wavelet, and the mean and standard deviation are calculated in terms of the subbands at each decomposed level. The decomposition procedure can be seen from Fig. 5.
- **Shape**. the edge direction histogram (EDH) [14] is employed to capture the spatial distribution of edges as a good shape figure. EDH is calculated upon the Y component in the YCrCb color space into five categories, namely, horizontal, 45 diagonal, vertical, 135 diagonal, and isotropic.

Each feature has its own power to characterize a type of image content. The system combines the color, texture, and shape features into a feature vector and then normalizes the vector into a normal distribution.

5 Experiments with Toy Problems

5.1 SVM is Unstable for a Small-Sized Training Set

The toy problem in Fig. 6 shows that the optimal hyperplane of the SVM is sensitive to small changes of the training set. The left figure shows an optimal hyperplane, which is trained by the original training set. The right figure shows a much different optimal hyperplane, which is trained by the original training set with only one incremental pattern.

5.2 SVM is Biased with Unbalanced Training Set

The toy problem in Fig. 7 shows that the optimal hyperplane of the SVM, which is trained by an unbalanced training set, will bias toward the class with more training samples. The left figure shows the overview of the training set. Through the right figure, which is cut from the bottom-right part of the left figure, we can see that the optimal hyperplane biases toward the class with more training examples.

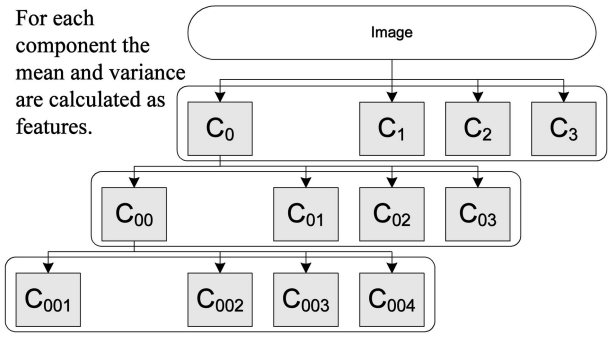


Fig. 5. Wavelet texture feature structure.

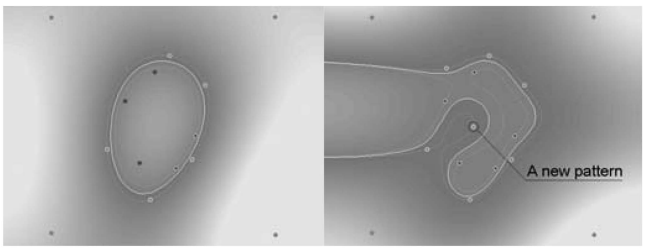


Fig. 6. SVM is unstable.

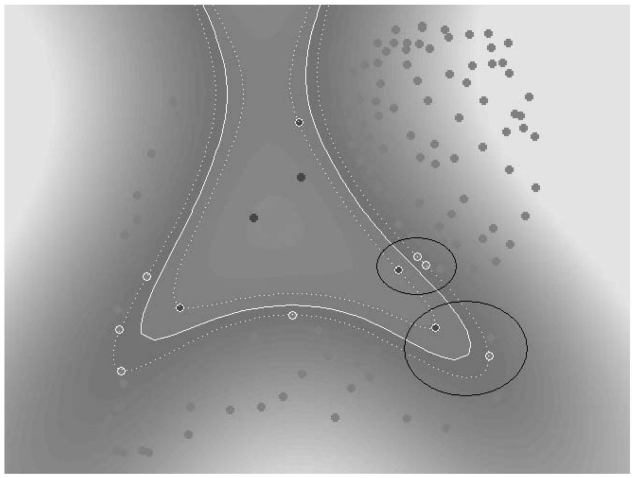


Fig. 7. SVM's optimal hyperplane is deflected.

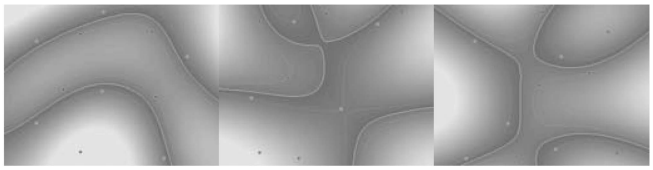


Fig. 8. The features are diverse.

Fig. 9. Samples in the relabeled image database.

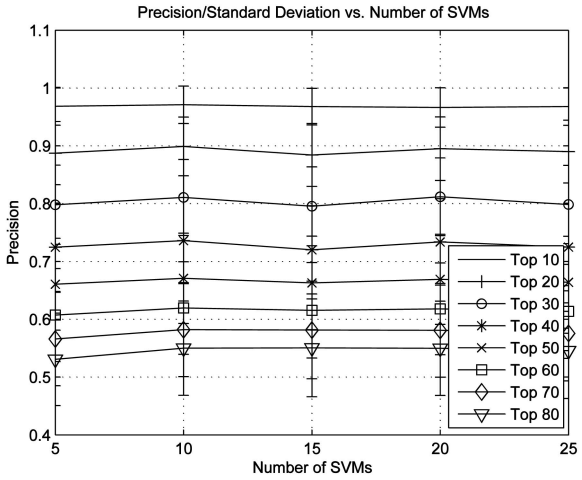


Fig. 10. AB-SVM-based relevance feedback.

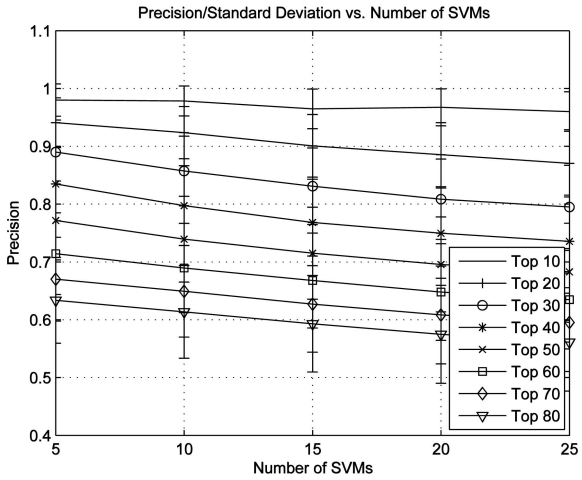


Fig. 11. RS-SVM-based relevance feedback.

5.3 The Visual Features Are Diverse for CBIR

This toy problem is constructed from some real-world data in relevance feedback. There are four positive and seven negative feedback samples. We randomly select two features to construct SVM's optimal hyperplane for three times. They are visualized in Fig. 8. We can see that the individual SVM classifiers are diverse with different features.

6 Experimental Results with Corel Images

Our image retrieval system is implemented with a real-world image database including 17,800 Corel images—a subset of the Corel Photo Gallery [30]. Some samples are shown in Fig. 9.

In the Corel Photo Gallery, each folder includes 100 images. However, the folders' names are not suitable as conceptual classes. So, the 17,800 images are manually labeled into about 90 concepts as the ground truth.

The experiments are simulated by a computer automatically. First, 300 queries are randomly selected from the data, and then relevance feedback is automatically done by the computer: All query relevant images (i.e., images of the same concept as the query) are marked as positive feedback samples in the top 40 images and all the other images are marked as negative feedback samples. In general, we have about five images as positive feedback samples. The procedure is close to real-world circumstances because the user typically would not like to click on the negative feedback samples.

Requiring the user to mark only the positive feedback samples in the top 40 images is reasonable. Since the visual features cannot well describe semantic contents, a system usually asks the user to mark three-four screen shots of images in the current retrieval process. Meanwhile, for some applications, the user would like to label only a small number of feedback samples and expect to get the best results hitherto wherever they terminate the query process. However, for some other users, they would like to cooperate with the system and, thus, can provide many (screens of) positive/negative examples [33]. In addition, even though the user stops earlier, the proposed scheme can still work well upon the Corel image database—when negative feedback samples are not as many as positive feedback samples, we can conduct bagging (not the asymmetric bagging) directly. Therefore, the assumption of top 40 images is made for our experiments below.

In this paper, precision and the standard deviation (SD) are used to evaluate the performance of a relevance feedback algorithm. Precision is the percentage of relevant images in the top N retrieved images. The SD serves as an error-bar, while the precision is the major evaluation method. As an important auxiliary of the precision, the SD can well describe the stability of different algorithms and, so, it is also a key

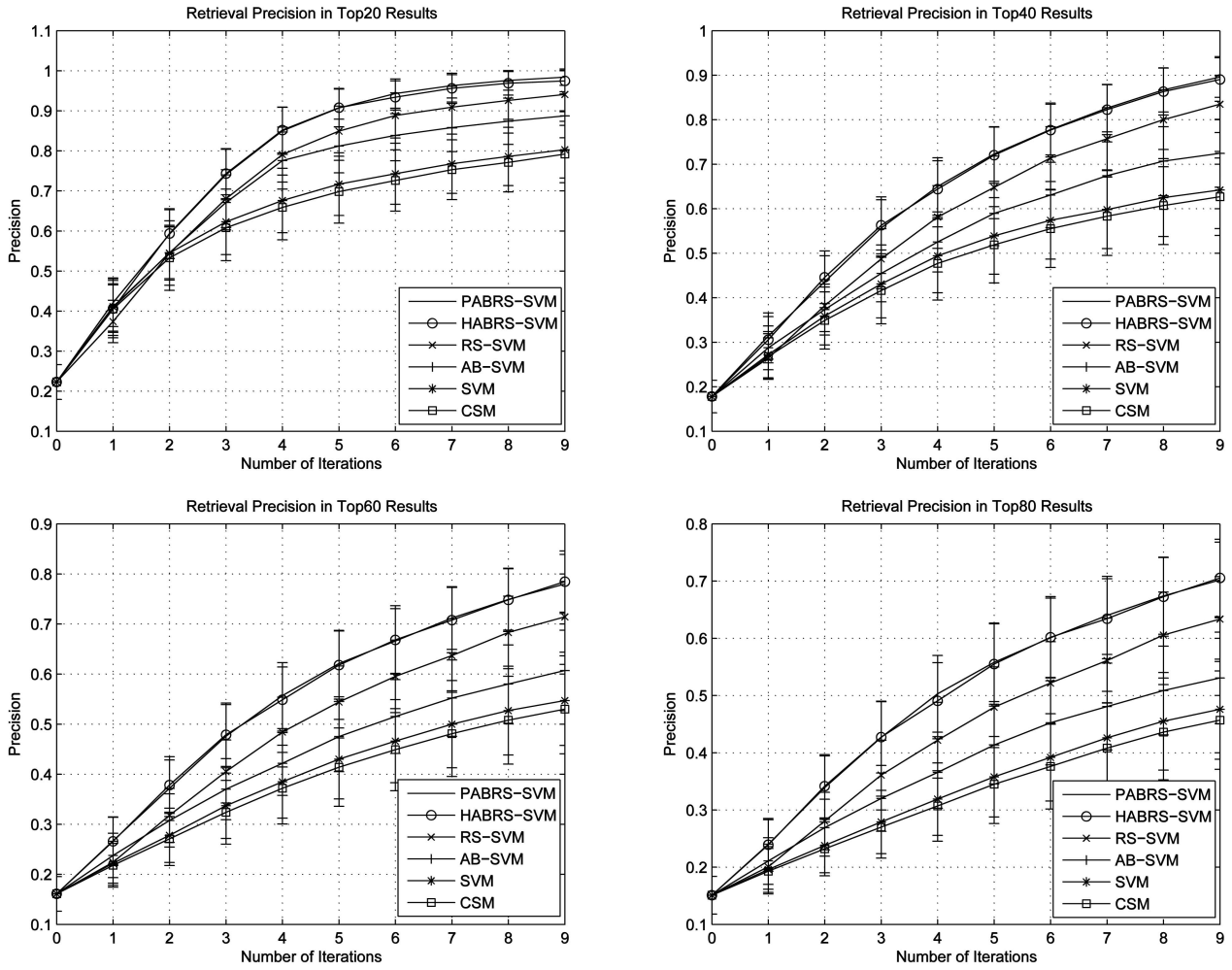


Fig. 12. Performance of all proposed algorithms compared to existing algorithms. The algorithms are evaluated over nine iterations.

feature of any relevance feedback schemes. The precision curve is the averaged precision values of 300 queries and the SD values are the standard deviation of the 300 queries' precision values. The precision curve evaluates the effectiveness of a given algorithm and the corresponding SD evaluates the robustness of the algorithm. In the precision and SD curves, 0 feedback refers to the retrieval based on the Euclidean distance measure without relevance feedback.

We compare all the proposed algorithms with the original SVM RF [31] and the constrained similarity measure using SVM (CSM)-based relevance feedback [6]. We chose the Gaussian kernel $K(\mathbf{x},\mathbf{y}) = e^{-\rho\|\mathbf{x}-\mathbf{y}\|^2}$ with $\rho = 1$ for all the algorithms because it achieves the best performances for all kernel-based algorithms according to a series of experiments with different kernel parameters. In our paper, we use the OSU-SVM [34] for all SVM-based RFs. Furthermore, since the Gaussian kernel is used here, the kernel Gram matrix is always full rank. Consequently, the training error is zero during the RF procedure for all SVM-based RFs.

6.1 Performance of AB-SVM

Fig. 10 shows the precision and SD values when using different numbers of SVMs in AB-SVM. The results demonstrate that the number of SVMs does not affect the performance of the asymmetric bagging method.

From these experiments, we can see that five weak SVMs are enough for AB-SVM, and AB-SVM clearly outperforms both SVM and CSM. The precision curve of AB-SVM is higher than that of SVM and CSM and the SD value is lower than that of SVM and CSM.

In Figs. 10, 11, 12, and 13, the SD (error bar) values are scaled by 4 for a clearer presentation.

6.2 Performance of RS-SVM

Fig. 11 shows the precision and SD values when using different numbers of SVMs of RS-SVM. The results show that the number of SVMs does not affect the performance of RS-SVM.

The results demonstrate that five weak SVMs are enough for the RS-SVM and RS-SVM outperforms SVM and CSM.

6.3 Performance of ABRS-SVM

This set of experiments evaluate the performances of the proposed ABRS-SVM, AB-SVM, and RS-SVM. We chose $T_s = 5$ for ABSVM, $T_f = 5$ for RS-SVM, and $T_s = T_f = 5$ for ABRS-SVM.

The results in Fig. 12 show that ABRS-SVM gives the best performance followed by RS-SVM then AB-SVM. They all outperform SVM and CSM.

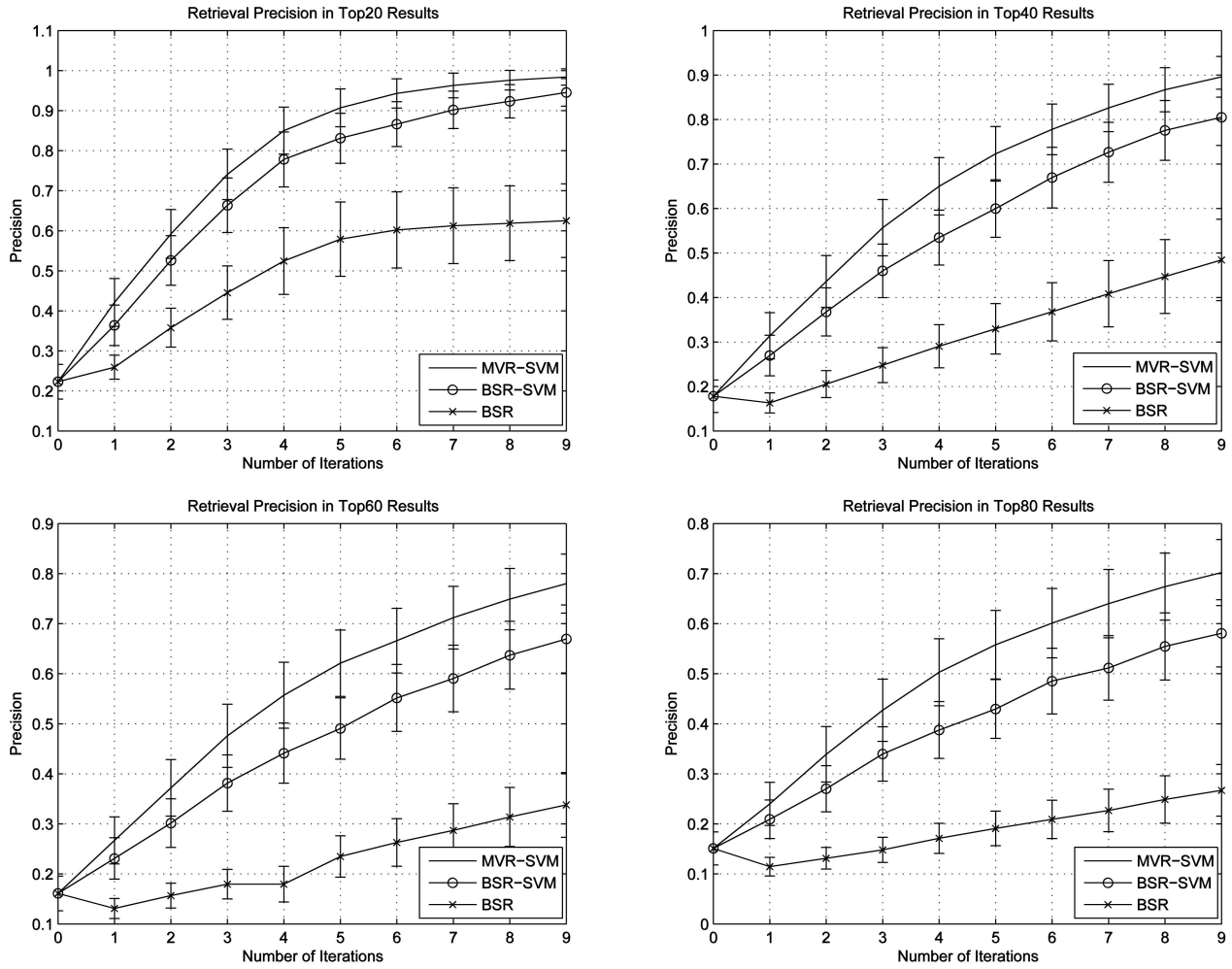


Fig. 13. Evaluation of aggregation models.

6.4 Evaluation of Aggregation Models

This set of experiments evaluates the performances of three different dissimilarity measures, MVR-SVM, BSR, and BSR-SVM, for the proposed ABRS-SVM-based relevance feedback. All algorithms are evaluated over nine iterations. The precision and SD curves are reported in Fig. 13. From this figure, we see that MVR-SVM can outperform both BSR-SVM and BSR.

BSR-SVM considers the behavior of weak classifiers, so, in theory, it may outperform MVR-SVM. However, according to the experimental results, its performance is worse than MVR-SVM, because we cannot estimate the behavior exactly for unstable weak classifiers. From Fig. 13, we find that BSR is much worse than MVR-SVM and BSR-SVM because MVR-SVM and BSR-SVM choose the best individual SVM to measure the dissimilarity between a given image and the user's sentiment. BSR uses the averaged probabilities, which cannot be estimated exactly. Therefore, MVR-SVM is the best choice.

6.5 Computational Complexity

Because the size of the training set is small, the overall computational complexity is mostly determined by the testing stage. The n_s for SVM RF is much bigger than that of AB-SVM and ABRS-SVM, and the n_f of SVM RF is much bigger than that of RS-SVM and ABRS-SVM. Thus, the computational complexity of SVM RF is much higher than that of AB-SVM, RS-SVM, and ABRS-SVM. In general, for each of the four algorithms, the inequality $n_f > n_s$ holds because the number of feedback samples is much smaller than the number of feature dimensions. Consequently, the computational complexity of AB-SVM is higher than RS-SVM and ABRS-SVM, and the computational complexity of RS-SVM is lower than that of ABRS-SVM. Our experiments have confirmed these observations.

To verify the efficiency of the proposed algorithms, we record the computational time in the experiments. The ratio for the time used by different methods are SVM: CSM: AB-SVM: RS-SVM: ABRS-SVM = 25: 25: 11: 3: 5. This shows that the new SVM-based algorithms are much more efficient than existing SVM-based algorithms.

7 Related Work

Our scheme in Section 4 is different from boosting in general and AdaBoost in particular. In Tieu and Viola's AdaBoost image retrieval [26], a set of weak two-class classifiers were incorporated by some weights and, thus, a strong classifier is built as a weighted sum of the weak classifiers. Our method

differs from both Tieu and Viola's AdaBoost image retrieval in [26] and Guo's AdaBoost [6], for the following reasons:

- Tieu and Viola employed more than 45,000 highly selective features. These features were demonstrated to be sparse with high kurtosis and were argued to be expressive for high-level semantic concepts [26], [33]. However, our scheme is designed to work upon the traditional low-level visual features (as introduced in Section 4). The constrained similarity measure (CSM) [6] claimed that the SVM outperforms AdaBoost when the traditional visual features are selected as the image description method.
- In [26], AdaBoost was used for feature selection based on the training set (the selected feedback samples). In our work, the new scheme randomly selects features and training samples.
- Boosting aims to minimize the training error by assigning a weight to each of a series of weak classifiers and then combining them for a vote. Conventional boosting schemes, such as AdaBoost, cannot boost the performance of strong classifiers (e.g., SVM) since the *zero-training-error* in strong classifiers is generally set as the criterion for stopping the boosting training. The *zero-training-error* is easy to get because the dimensionality of the low-level features is much larger than the number of the feedback samples in relevance feedback. Consequently, boosting methods degenerate to a single strong classifier. On the contrary, for our presented scheme, from the mathematical description of the working principle, we can see that the scheme can combine a series of diverse SVM classifiers and yield better performances.
- Both SVM and Adaboost use the margin maximization criterion (MMC). However, SVM and AdaBoost use L_2 norm and L_1 norm for weight vectors, respectively. Furthermore, SVM and AdaBoost use L_2 norm and $L_{\inf}$ norm for instance vectors, respectively.

There are also significant differences between boosting and the bagging approach we have used. First, boosting trains weak classifiers based on a resampling using the training error information, so boosting can only improve the performance of basic classifiers, which cannot correctly classify some of the training samples. In relevance feedback, the samples' size is smaller than the number of the samples' feature dimensions, so boosting cannot work well in our context when the selected basic classifiers are SVMs. Second, for relevance feedback in retrieval, the feedback samples are limited, especially the positive feedback samples, so the weights for boosting-leant basic/weak classifiers are only optimal for the training set but not the testing set. Unlike the boosting, bagging does not generate any weights for its weak classifiers. Therefore, bagging does not meet the overfitting issues in boosting.

Furthermore, we do not use traditional supervised feature selection methods because they generally face an overfitting problem as well. This overfitting problem is due to the limited feedback samples especially the positive samples in the relevance feedback procedure. Unlike traditional supervised feature selection methods, the random subspace method does not select features to minimize the training error or maximize the margin between the positive and negative feedback samples, but utilizes the diverse characteristics of different features. The mathematical explanation of the working principle of the framework in Section 3.2 can well explain this claim.

For unsupervised feature selection, it is always based on some selection criterion. When the criterion is good enough to capture the user's preferences, it might work more effectively than ours. However, because the size of a real-world image database can be very large, to find such a criterion seems to be impossible. In addition, different users can have different viewpoints on the same images. Finally, unsupervised feature selection is time consuming and, thus, it does not fit with online learning. Consequently, using the same unsupervised feature selection criterion for all users is not a wise option.

8 Conclusion

With the explosive growth in image records and the rapid increase of computer power, retrieving images from a large-scale image database becomes one of the most active research fields. Content-based image retrieval (CBIR) is a technique to retrieve images semantically relevant to the user's query from an image database. Relevance feedback (RF) is a way of bridging this gap and scaling the performance in CBIR systems. Support vector machines (SVM) based relevance feedback is widely employed. However, when the number of positive feedback samples is small, the performance of SVM RF becomes poor because of the following facts: 1) the SVM classifier is unstable for a small-sized training set, 2) the optimal hyperplane of SVM may be biased, and 3) SVM always encounters an overfitting problem. In this paper, we have designed a new *asymmetric bagging and random subspace mechanism* (with three algorithms) to address the three key problems. Extensive experiments on a Corel Photo database with 17,800 images have shown that the new mechanism can improve the performance of relevance feedback significantly.

In our proposed system, it is found that the kernel parameters affect the performance of relevance feedback algorithms. Unfortunately, how to select the kernel parameters is a problematic issue. Recently, a tuning method has been provided in our system for the user to select the parameters of SVM. In the future, the tuning method will be tested and generalized to select the parameters of kernel-based algorithms. For relevance feedback in CBIR, the size of the training set is small, so the *leave-one-out* method to tune the parameters will be a good choice.

Acknowledgments

The authors would like to thank the four anonymous reviewers for their constructive comments on the first two versions of this paper. The research was supported by the Research Grants Council of the Hong Kong SAR (under project number AoE/E-01/99).

REFERENCES

[1] D. Bahler and L. Navarro, "Methods for Combining Heterogeneous Sets of Classifiers," *Proc. 17th Nat'l Conf. Am. Assoc. for Artificial Intelligence,* 2000.

[2] L. Breiman, "Bagging Predictors," *Machine Learning,* vol. 24, no. 2, pp. 123-140, 1996.

[3] 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.

[4] T. Chang and C. Kuo, "Texture Analysis and Classification with Tree-Structured Wavelet Transform," *IEEE Trans. Image Processing,* vol. 2, no. 4, pp. 429-441, 1993.

[5] Y. Chen, X. Zhou, and T.S. Huang, "One-Class SVM for Learning in Image Retrieval," *Proc. IEEE Int'l Conf. Image Processing,* pp. 815-818, 2001.

[6] G. Guo, A.K. Jain, W. Ma, and H. Zhang, "Learning Similarity Measure for Natural Image Retrieval with Relevance Feedback," *IEEE Trans. Neural Networks,* vol. 12, no. 4, pp. 811-820, 2002.

[7] T.K. Ho, "The Random Subspace Method for Constructing Decision Forests," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 8, pp. 832-844, Aug. 1998.

[8] P. Hong, Q. Tian, and T.S. Huang, "Incorporate Support Vector Machines to Content-Based Image Retrieval with Relevant Feedback," *Proc. IEEE Int'l Conf. Image Processing,* pp. 750-753, 2000.

[9] A. Jain and A. Vailaya, "Image Retrieval Using Color and Shape," *Pattern Recognition,* vol. 29, no. 8, pp. 1233-1244, 1996.

[10] M. Jordan and R. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm," *Int'l. J. Neural Computation,* vol. 6, no. 5, pp. 181-214, 1994.

[11] D. Kim and C. Kim, "Forecasting Time Series with Genetic Fuzzy Predictor Ensemble," *IEEE Trans. Fuzzy Systems,* vol. 5, no. 4, pp. 523-535, 1997.

[12] J. Kittler, M. Hatef, P.W. Duin, and J. Matas, "On Combining Classifiers," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 3, pp. 226-239, Mar. 1998.

[13] B. Manjunath and W. Ma, "Texture Features for Browsing and Retrieval of Image Data," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 8, pp. 837-842, Aug. 1996.

[14] B. Manjunath, J. Ohm, V. Vasudevan, and A. Yamada, "Color and Texture Descriptors," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 11, no. 6, pp. 703-715, 2001.

[15] J. Mao and A. Jain, "Texture Classification and Segmentation Using Multiresolution Simultaneous Autoregressive Models," *Pattern Recognition,* vol. 25, no. 2, pp. 173-188, 1992.

[16] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubino, "The QBIC Project: Querying Images by Content Using Color, Texture, and Shape," *Proc. SPIE Storage and Retrieval for Images and Video Databases,* pp. 173-181, 1993.

[17] G. Pass, R. Zabih, and J. Miller, "Comparing Images Using Color Coherence Vectors," *Proc. ACM Int'l Conf. Multimedia,* pp. 65-73, 1996.

[18] J. Peng, "MultiClass Relevance Feedback Content-Based Image Retrieval," *Computer Vision and Image Understanding,* vol. 90, no. 1, pp. 42-67, 2003.

[19] J. Platt, "Probabilistic Outputs for Support Vector Machines and Comparison to Regularized Likelihood Methods," *Proc. Advances in Large Margin Classifiers,* pp. 61-74, 2000.

[20] G. Ratsch, S. Mika, B. Scholkopf, and K.R. Muller, "Constructing Boosting Algorithms from SVMs: An Application to One-Class Classification," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 9, pp. 1184-1199, Sept. 2002.

[21] Y. Rui, T.S. Huang, and S. Mehrotra, "Content-Based Image Retrieval with Relevance Feedback in MARS," *Proc. IEEE Int'l Conf. Image Processing,* vol. 2, pp. 815-818, 1997.

[22] 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.

[23] M. Swain and D. Ballard, "Color Indexing," *Int'l J. Computer Vision,* vol. 7, no. 1, pp. 11-32, 1991.

[24] H. Tamura, S. Mori, and T. Yamawaki, "Texture Features Corresponding to Visual Perception," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 8, no. 6, pp. 460-473, 1978.

[25] D. Tao and X. Tang, "Random Sampling Based SVM for Relevance Feedback Image Retrieval," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* pp. 647-652, 2004.

[26] K. Tieu and P. Viola, "Boosting Image Retrieval," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 228-235, 2001.

[27] S. Tong and E. Chang, "Support Vector Machine Active Learning for Image Retrieval," *Proc. ACM Int'l Conf. Multimedia,* pp. 107-118, 2001.

[28] V. Tresp and M. Taniguchi, "Combining Estimators Using Non-Constant Weighting Functions," *Advances in Neural Information Processing Systems,* pp. 419-426, 1995.

[29] V. Vapnik, *The Nature of Statistical Learning Theory.* Springer-Verlag, 1995.

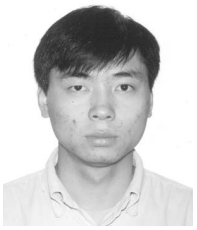
[30] J. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 9, pp. 947-963, Sept. 2001.

[31] L. Zhang, F. Lin, and B. Zhang, "Support Vector Machine Learning for Image Retrieval," *Proc. IEEE Int'l Conf. Image Processing,* pp. 721-724, 2001.

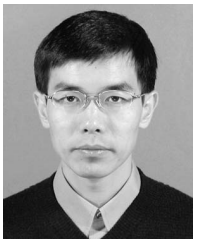
[32] X. Zhou and T.S. Huang, "Small Sample Learning During Multimedia Retrieval Using Biasmap," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 11-17, 2001.

[33] X. Zhou and T.S. Huang, "Relevance Feedback for Image Retrieval: A Comprehensive Review," *ACM Multimedia Systems J.,* vol. 8, no. 6, pp. 536-544, 2003.

[34] http://www.ece.osu.edu/~maj/osu_svm/, 2002.



Dacheng Tao received the BEng degree from the University of Science and Technology of China and the MPhil degree from the Chinese University of Hong Kong. He is currently a PhD candidate at the University of London. His research interests include acoustic signal matching and retrieval, bioinformatics, biometrics, computer vision, data mining, image processing, machine learning, multimedia information retrieval, pattern classification, and visual surveillance. He has published extensively with *IEEE TPAMI*, *TIP*, *TMM*, *TCSVT*, CVPR, ACM SIGMM, etc. He was a program committee member for several conferences such as the Sixth IEEE International Workshop on Visual Surveillance (in conjunction with ECCV 2006). Previously he gained several Meritorious Awards from the International Interdisciplinary Contest in Modeling, which is the highest level mathematical modeling contest in the world, organized by COMAP. He is a reviewer for a number of major journals and conferences such as *TIP*. He is a student member of the IEEE.



Xiaoou Tang received the BS degree from the University of Science and Technology of China, Hefei, in 1990, and the MS degree from the University of Rochester, Rochester, New York, in 1991. He received the PhD degree from the Massachusetts Institute of Technology, Cambridge, in 1996. He was a professor and the director of Multimedia Lab in the Department of Information Engineering, the Chinese University of Hong Kong until 2005. Currently, he is the group manager of the Visual Computing Group at Microsoft Research Asia. He is the local chair of the IEEE International Conference on Computer Vision (ICCV) 2005, the area chair of ICCV '07, the program cochair of ICCV '09, and the general cochair of the IEEE ICCV International Workshop on Analysis and Modeling of Faces and Gestures 2005. He is a guest editor of the special issue on underwater image and video processing for *IEEE Journal of Oceanic Engineering* and the special issue on image and video-based biometrics for *IEEE Transactions on Circuits and Systems for Video Technology*. He is an associate editor of *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*TPAMI*). His research interests include computer vision, pattern recognition, and video processing. He is a senior member of the IEEE and the IEEE Computer Society.

Xuelong Li is a lecturer at the University of London. His current research interests include visual surveillance, biometrics, data mining, multimedia information retrieval, cognitive computing, image processing, pattern recognition, and industrial applications. He has published more than 40 papers in the journals (*TCSVT*, *TIP*, *TMM*, *TPAMI*, etc.) and conferences (CVPR, ICASSP, ICDM, etc.). He was on the PCs of more than 30 conferences (BMVC, ECIR, ICDM, ICME, ICMLC, ICPR, PAKDD, PCM, SMC, WI, etc.). He has cochaired the fifth UKCI. He is an AE of the *IJIG* (World Scientific) and *Neurocomputing* (Elsevier). He is also on the EB of *IJITDM* (World Scientific). He is a GE for an *IJCM* (Taylor & Francis) special issue. He is a reviewer for around 60 journals and conferences, including nine IEEE transactions. He is a member of the IEEE.

Xindong Wu received the PhD degree in artificial intelligence from the University of Edinburgh, Britain. He is a professor and the chair of the Department of Computer Science at the University of Vermont. His research interests include data mining, knowledge-based systems, and Web information exploration. He has published extensively in these areas in various journals and conferences, including *IEEE TKDE*, *TPAMI*, *ACM TOIS*, *IJCAI*, *AAAI*, *ICML*, *KDD*, *ICDM*, and *WWW*, as well as 12 books and conference proceedings. Dr. Wu is the Editor-in-Chief of the *IEEE Transactions on Knowledge and Data Engineering*, the founder and current steering committee chair of the IEEE International Conference on Data Mining (ICDM), an honorary Editor-in-Chief of *Knowledge and Information Systems* (Springer), and a series editor of the Springer book series on advanced information and knowledge processing (AI&KP). He is the 2004 ACM SIGKDD Service Award winner. He is a senior member of the IEEE and the IEEE Computer Society.