Mean Version Space: a New Active Learning Method for Content-Based Image Retrieval*

Jingrui He[1], Mingjing Li[2], Hong-Jiang Zhang[2], Hanghang Tong[1], Changshui Zhang[3]
[1,3]Department of Automation, Tsinghua University, Beijing 100084, China
[2]Microsft Research Asia, 49 Zhichun Road, Beijing 100080, China
[1]{hejingrui98, walkstar98}@mails.tsinghua.edu.cn, [2]{mjli, hjzhang}@microsoft.com
[3]zcs@tsinghua.edu.cn

ABSTRACT

In content-based image retrieval, relevance feedback has been introduced to narrow the gap between low-level image feature and high-level semantic concept. Furthermore, to speed up the convergence to the query concept, several active learning methods have been proposed instead of random sampling to select images for labeling by the user. In this paper, we propose a novel active learning method named mean version space, aiming to select the optimal image in each round of relevance feedback. Firstly, by diving into the lemma that motivates support vector machine active learning method ($\text{SVM}_{\text{active}}$), we come up with a new criterion which is tailored for each specific learning task and will lead to the fastest shrinkage of the version space in all cases. The criterion takes both the size of the version space and the posterior probabilities into consideration, while existing methods are only based on one of them. Moreover, although our criterion is designed for SVM, it can be justified in a general framework. Secondly, to reduce processing time, we design two schemes to construct a small candidate set and evaluate the criterion for images in the set instead of all the unlabeled images. Systematic experimental results demonstrate the superiority of our method over existing active learning methods.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications – *image databases*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *relevance feedback, search process.*

General Terms

Algorithms, Experimentation, and Theory.

Keywords

Content-based image retrieval, relevance feedback, active learning, version space

1. INTRODUCTION

The last few decades witnessed an explosion in the volume of digital images, which necessitates an efficient scheme for browsing and indexing large image databases. To address this issue, people have proposed an integrated framework named content-based image retrieval (CBIR). In the framework, firstly, each image is mapped to a point in the feature space by extracting low-level image features, which can be categorized into color [2, 7, 14], texture [3, 4, 19], shape [11, 12, 23], etc; secondly, given a query in terms of image examples, the framework then retrieves images based on their features.

It is widely accepted that the major bottleneck of CBIR systems is the large gap between low-level image features and high-level semantic concepts, which prevents the systems from being applied to real applications [10]. To be specific, images of dissimilar semantic content may share some common low-level features, while images of similar semantic content may be scattered in the feature space. Despite the great deal of research work dedicated to the exploration of an ideal descriptor for image content, no single feature or feature combination can achieve satisfactory performance up till now.

To narrow or bridge the gap, relevance feedback, an efficient online learning technique borrowed from the field of information retrieval, has been introduced to CBIR since the 1990's [10]. In each round of relevance feedback, the user will judge the relevance of some database images, and the system will update its retrieval result according to these newly obtained labeled examples. Two most important factors in relevance feedback are the image selection strategy and the learning method [10, 16].

The learning method in relevance feedback has been extensively studied. Traditional learning methods can be categorized into three major groups [9]: query reweighting, query point movement, and query expansion. However, because these methods do not fully utilize the information embedded in feedback images, their performance is far from satisfactory. More recently, statistical learning methods have been applied to relevance feedback. Among others, some researchers apply inductive methods to the learning task, aiming to create a classifier that generalizes well on unseen examples. For example, the authors of [15] first compute a large number of highly selective features, and then use boosting to learn a classification function in this feature space; similarly, the

* This work was performed at Microsoft Research Asia.

learning method proposed in [22] trains a support vector machine (SVM) from labeled examples, hoping to obtain a small generalization error by maximizing the margin between the two classes of images. On the other hand, some researchers consider image retrieval as a transductive learning problem, aiming to accurately predict the relevance of unlabeled images attainable during the training stage. For example, the authors of [21] propose a discriminant-EM algorithm. It makes use of unlabeled data to construct a generative model, which will be used to measure relevance between the query and database images.

In contrast, there is not so much work dealing with image selection strategy. A good strategy should select the most informative images in each round of relevance feedback, thus the user only need to label a small number of images before the system learns the query concept. The simplest selection strategy is random sampling. However, since the selected images to be labeled by the user often convey little information for improving present retrieval result, a large number of images have to be labeled before the system achieves satisfactory performance, which runs counter to our original intention [16]. A more reasonable choice is to select the most relevant images, the motivation behind which is to ask the user to validate the judgment of the current system on image relevance. However, from the classification point of view, these images may not be sufficient to train an accurate classifier. More recently, the authors of [16, 17] propose a support vector machine active learning method as the image selection strategy, and prove that, for a given number of queries, it minimizes the maximum expected size of the version space, where the maximum is taken over all conditional distributions of image label given its low-level feature. However, in practice, people have found that SVM$_{\text{active}}$ often leads to slow convergence to the target concept [20], and its performance is sometimes not as good as that of the most relevant strategy. Another example is the active learning method using pre-clustering proposed in [6], which takes into account the prior data distribution. It makes two assumptions: (1) all the clusters have the shape of hyper-spheres in the feature space; (2) given the cluster label, the data and its class label are independent. However, in the context of image retrieval, both of them may be violated due to the complex distribution of low-level image features. Furthermore, the method is restricted to linear logistic regression to model the distribution of cluster labels, which may not be powerful enough for image retrieval tasks.

In this paper, we aim to improve SVM$_{\text{active}}$ from a theoretical point of view. Firstly, by diving into the lemma that motivates SVM$_{\text{active}}$, we draw a conclusion that the inferiority of SVM$_{\text{active}}$ may be attributed to the fact that it considers the supremum of all learning tasks, while ignoring their specialty. Then we propose a new criterion for active learning which is tailored for each specific learning problem. The criterion guarantees that after selecting a single image, the mean version space will be maximally shrinked. As we will show in Section 2, the criterion takes both the size of the version space and the posterior probabilities into consideration, while SVM$_{\text{active}}$ and the most relevant strategy are only based on one of them. Our criterion is justified with both unbiased and biased SVM classifiers. Moreover, although our criterion is designed for SVM, it can be justified in a general framework. Secondly, since the evaluation of the proposed criterion over the entire database may be very time-consuming, and the response time of a CBIR system is of key importance, we propose two simple schemes to limit the candidate images. Systematic experiments on a general-purpose image database consisting of 5,000 Corel images validate the superiority of our method over existing ones.

The paper is organized as follows. In Section 2, we analyze the limitation of SVM$_{\text{active}}$, propose our mean version space criterion for active learning in detail, and examine it under a general framework. The two schemes for speeding up the evaluation process are presented in Section 3. We provide systematic experimental results to evaluate the proposed active learning method from various aspects in Section 4. Finally, we conclude the paper in Section 5.

2. MEAN VERSION SPACE ACTIVE LEARNING

2.1 Notations and Assumptions

In this paper, we adopt SVM [18] as the learning method in relevance feedback. Given a set of labeled examples $x_k,\ k=1,\ldots,M,\ x_k\in R^N$, and their labels $y_k,\ k=1,\ldots,M,\ y_k\in\{-1,\ 1\}$ (M is the total number of labeled examples), a Mercer kernel K implicitly defines a mapping $\Phi: R^N \to F$. In the feature space F, there are a set of hyperplanes that separate the examples. These hyperplanes are called the version space [5], which is formally defined as follows [17]:

DEFINITION 1. The version space

$$V=\left\{w\in W \,\middle|\, \|w\|=1,\ y_k\left(w\cdot\Phi(x_k)\right)>0,\ \mathrm{k}=1,\ldots,M\right\} \qquad (1)$$

where the parameter space W is equal to F.

In the version space, SVM selects the hyperplane that maximizes the margin in F, i.e. the optimal parameter

$$w^* = \underset{w\in W}{\arg\max}\ \min_k\left\{y_k\left(w\cdot\Phi(x_k)\right)\right\} \qquad (2)$$

Here we only consider unbiased SVM classifier, i.e., the optimal separating hyperplane is chosen from the hyperplanes that pass through the origin. Intuitively, this limitation might bring about performance degradation since it only covers a small subset of all possible hyperplanes that separate the examples. We will come back to this issue in subsection 2.3.

Let $Area(V)$ denote the surface area that the version space V occupies on the hypersphere $\|w\|=1$ [17]. Given an active learner l, let V_i denote the version space of l after i queries have been made, while $V_i^+(x_k)$ and $V_i^-(x_k)$ denote the version space after the $(i+1)th$ query x_k is labeled as 1 and -1 respectively.

Note that the version space exists only if the training examples are linearly separable in the feature space F. Since the number of labeled examples is usually very small compared with the dimensionality of F, we make a reasonable assumption that the relevant and irrelevant images previously marked by the user can be separated by a hyperplane in F. As in [17], we also require that the examples have constant modulus in F, i.e. $\|\Phi(x_k)\|=\lambda$.

This requirement has no effect on radial basis function (RBF) kernel $K(u,v)=\exp\left(-\|u-v\|^2/\sigma^2\right)$; while for polynomial kernel $K(u,v)=(u\cdot v+1)^p$, it requires that $\|x_k\|$ be constant.

2.2 Mean Version Space

The lemma that motivates SVM$_{\text{active}}$ is as follows [17]:

LEMMA 1. Suppose we have finite dimensional feature space F. Suppose active learner l^* always queries instances whose corresponding hyperplanes in parameter space W halve the area of the current version space. Let l be any other active learner. Denote the version spaces of l^* and l after i queries as V_i^* and V_i respectively. Let P denote the set of all conditional distributions of y given x. Then

$$\forall i\in\mathbb{N}^+\ \sup_{P\in\mathrm{P}} E_P\left[Area\left(V_i^*\right)\right]\le\sup_{P\in\mathrm{P}} E_P\left[Area\left(V_i\right)\right] \tag{3}$$

with strict inequality whenever there exists a query $j\in\{1,\ldots,i\}$ by l that does not halve version space V_{j-1}.

The lemma implicitly assumes that the criterion for selecting examples in SVM$_{\text{active}}$ is:

$$c_{SVM}(x_k)=\left|Area\left(V_i^+(x_k)\right)-Area\left(V_i^-(x_k)\right)\right| \tag{4}$$

SVM$_{\text{active}}$ then selects the example with the smallest c_{SVM} in each round of relevance feedback. Due to practical difficulty in computing the size of the version space, the method finally takes on the closest-to-boundary criterion to select examples [16, 17]. Note that according to ***Lemma 1***, SVM$_{\text{active}}$ is optimal in the supremum sense. However, for a specific learning task where P is fixed, the above lemma cannot guarantee that SVM$_{\text{active}}$ maximally shrinks the version space. This may partially explain why SVM$_{\text{active}}$ often brings about slow convergence to the target concept, and its performance is sometimes not as good as that of the most relevant strategy.

Before presenting our own criterion for selecting examples, we would like to reiterate the goal for designing an active learning strategy combined with SVM, i.e., to maximally shrink the version space after each selected example is labeled by the user, since a small version space will guarantee that the predicted hyperplane lies close to the optimal one constructed when all the images have their labels. Quite naturally, we define the criterion as the expectation of the size of the version space after an unlabeled example x_k has been labeled. To be specific,

$$\begin{aligned}c_{MVS}(x_k)=Area\left(V_i^+(x_k)\right)P\left(y_k=1|x_k\right)+\\ Area\left(V_i^-(x_k)\right)P\left(y_k=-1|x_k\right)\end{aligned} \tag{5}$$

If we select example x^* which has the smallest c_{MVS}, the version space will be maximally shrinked with each round of relevance feedback. Note that our criterion is tailored for each specific learning task by considering the posterior probabilities ($P(y_k=1|x_k)$ and $P(y_k=-1|x_k)$), in contrast to SVM$_{\text{active}}$, which ignores this information and is optimal only in the supremum sense.

To obtain the criterion for each unlabeled example, we need to calculate both the size of version space $V_i^+(x_k)$, $V_i^-(x_k)$ and the posterior probabilities. It has been pointed out in [17] that it is not practical to explicitly compute the former term. Since there exists a duality between the feature space F and the parameter space W, and the examples have constant modulus in F (the assumption mentioned in subsection 2.1), a reasonable way to approximate $V_i^+(x_k)$ and $V_i^-(x_k)$ is as follows [17]: add x_k to the positive example set, retrain SVM to obtain its margin $m^+(x_k)$, which is used as an indication of the size of $V_i^+(x_k)$; add x_k to the negative example set, retrain SVM to obtain its margin $m^-(x_k)$, which is used as an indication of the size of $V_i^-(x_k)$.

On the other hand, the calculation of the posterior probabilities should be based on SVM trained on present labeled examples, which provides an estimation of the true probabilities. Since SVM outputs uncalibrated values, we need to convert the outputs to probabilities. In [8], the authors describe an intuitive way for such conversion, i.e.

$$P(y=1|f)=\frac{1}{1+\exp(Af+B)} \tag{6}$$

where f is the output of SVM, while A and B are real-valued parameters determined by maximum likelihood estimation. As long as $A<0$, $P(y=1|f)$ is an increasing function of f, which is consistent with our intuition. When the number of labeled images used for fitting this sigmoid function is very small, we may add some top-ranked (say, the first 20) images to the positive set and some bottom-ranked (say, the last 20) images to the negative set.

When an example is far from the boundary, i.e., it has a large value for $|f|$, if it is selected and assigned with the label predicted by the current classifier, the version space will shrink a little; if it is assigned with the opposite label, the size of the version space will be greatly reduced. However, since $|f|$ is large, its posterior probability of having the opposite label will be very small, which causes its criterion value to be large. On the other hand, when an example is within the margin, if it is selected and marked by the user, no matter what label it obtains, the version space will be reduced considerably, and the criterion value tends to be small. However, there is no guarantee that the examples on or near the separating hyperplane will be selected with the smallest criterion value, as in SVM$_{\text{active}}$, which reflects the difference between the two methods.

At the end of this subsection, we would like to derive the criterion of the most relevant strategy for theoretical comparison among the three image selection strategies:

$$c_{MR}=-f \tag{7}$$

The example with the smallest c_{MR} will be selected in each round of relevance feedback. As aforementioned, the posterior probabilities are determined by f. Therefore, this criterion can be considered as a function of the posterior probabilities. Note that the criterion of SVM$_{\text{active}}$ is only based on the size of the version space. Nevertheless, the mean version space criterion takes both the two factors into account, thus may obtain a better performance than these two image selection criterions.

2.3 Biased SVM Classifier

Up till now, we have focused on unbiased SVM classifier, i.e.,

$$f(x) = w \cdot \Phi(x) \tag{8}$$

However, as discussed in subsection 2.1, its performance might not be as good as that of biased SVM classifier, which can be written as

$$f(x) = w \cdot \Phi(x) + b \tag{9}$$

Note that its parameter space W' has one more dimension than W incurred by b, the translation factor. Therefore, the duality no longer exists between the feature space F and the parameter space W', which adds difficulties to the analysis of the version space. However, we can still use Equation 5 as the criterion to select examples in each round of relevance feedback except that the terms $Area\left(V_i^+(x_k)\right)$ and $Area\left(V_i^-(x_k)\right)$ must be replaced by the margins $m^+(x_k)$ and $m^-(x_k)$ respectively, since the concept of the version space is not considered in this case, i.e.

$$\begin{aligned} c_{MVS}(x_k) &= m^+(x_k) P(y_k = 1|x_k) \\ &\quad + m^-(x_k) P(y_k = -1|x_k) \end{aligned} \tag{10}$$

The reason for applying the above criterion to biased SVM classifier can be explained as follows: when the examples are linearly separable in the feature space F, labeling an example and adding it to the training set will cause the margin to decrease or to remain the same; when we finally obtain the ideal separating hyperplane, its margin will be the smallest in the entire training process. From this point of view, the most efficient way of selecting examples in each round of relevance feedback would be to select the ones which will maximally reduce the margin.

2.4 Mean Version Space Active Learning in a General Framework

It is widely accepted that active learning should select examples that minimize the expected future classification error [1]:

$$\int_x E\left[(\hat{y} - y)^2 | x\right] p(x)\,dx \tag{11}$$

where y is the true label of x, $\hat{y}$ is the label predicted by the updated classifier, and $E\left[.|x\right]$ denotes the expectation over $P(y|x)$.

Equation 11 can be considered as the generalization error of the updated classifier. In a binary classification problem, given an unlabeled example x_k, the expected generalization error after x_k is labeled by the user can be expressed as follows:

$$\begin{aligned} Error(x_k) &= \int_x E\left[(\hat{y}^+ - y)^2 | x\right] p(x)\,dx\; P(y_k = 1|x_k) \\ &\quad + \int_x E\left[(\hat{y}^- - y)^2 | x\right] p(x)\,dx\; P(y_k = -1|x_k) \end{aligned} \tag{12}$$

where $\hat{y}^+$ and $\hat{y}^-$ denote the predicted label after x_k has been labeled 1 and -1 respectively. Accordingly, $\int_x E\left[(\hat{y}^+ - y)^2 | x\right] p(x)\,dx$ and $\int_x E\left[(\hat{y}^- - y)^2 | x\right] p(x)\,dx$ are the generalization errors in those two cases.

When we use SVM, the generalization error can be evaluated by means of the margin size: the smaller the margin is, the less possible the constructed classifier with the newly labeled example will make a mistake when predicting the label of an unlabeled example. This argument seems to be contradictory with our common knowledge that a hyperplane with a large margin tends to have a smaller generalization error. The difference here is that we compare the different OPTIMAL hyperplanes obtained using different training sets. Note that a small margin means that the current hyperplane is close to the ideal one, thus the generalization error tends to be small accordingly. This conclusion is consistent with [13] in which a "best worst-case" model is used to induce the closest-to-boundary criterion for active learning. By replacing the two generalization error terms in Equation 12 by $m^+(x_k)$ and $m^-(x_k)$, we obtain Equation 10.

3. ACCELERATING THE EVALUATION PROCESS

One major problem with the proposed mean version space active learning method is that evaluating each unlabeled example requires solving two quadratic programming (QP) problems. When there are a lot of unlabeled examples, say many thousands, the processing time for selecting even a single one for the user to label is unbearable. Therefore, we need to design an efficient as well as effective acceleration scheme.

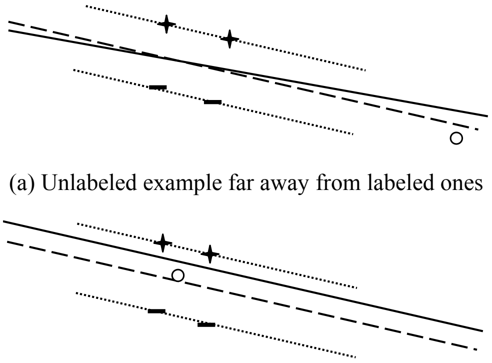
One way to reduce the processing time is to limit the scope of candidate examples. As has been discovered by the authors of [13], a point's location with respect to the labeled examples has a large effect on how labeling it influences the hyperplane. Here we use a similar example as in [13] to explain our idea, which is illustrated in Figure 1. As explained in subsection 2.2, the examples within the margin tend to have a small criterion value. Here, we assume that the selected examples lie within the margin for simplicity.

As we can see from Figure 1, an unlabeled example in the near neighborhood of labeled ones tends to change the separating hyperplane greatly, while an unlabeled example far away from any labeled example will bring little change to the placement of the hyperplane. Furthermore, in the context of CBIR, the number of labeled images is very small in most cases. Thus the estimated posterior probabilities with respect to the relevance to the user's query concept is relatively more accurate for examples in the neighborhood of labeled ones than those far away from any labeled example.

Based on the above two observations, we limit the scope of candidate examples to those nearest to the labeled ones. However, in practice, it is hard to define the nearest neighbors with respect to the labeled set, and the nearest neighbor search itself might be time-consuming. To address this issue, we design two simple schemes to construct the candidate set:

- Using the S most relevant images;

- Using the S images closest to the separating hyperplane.



(a) Unlabeled example far away from labeled ones

(b) Unlabeled example near labeled ones

Figure 1. The location of a selected example will largely affect the change in the separating hyperplane. "+" denotes a positive example, "-" denotes a negative example, and "o" denotes the selected unlabeled example. The dashed line is the old separating hyperplane; the solid line is the new hyperplane if the unlabeled example is given label -1; the area between the two dotted lines is the margin.

The rationality of the two schemes can be explained as follows: the first scheme selects the examples that are close to the positive ones, while the second scheme selects the examples that are close to the support vectors. On the other hand, the two schemes can be considered as using SVM$_{\text{active}}$ and the most relevant strategy for rough selection, followed by the mean version space criterion for further improvement.

To choose a proper value for S, we first sort all the unlabeled images according to the first/second scheme, and then calculate C_{MVS} for the images one by one. The evaluation process stops if one of the following conditions is satisfied:

- C_{MVS} of the present image is twice as large as the minimum value that has ever been reached;
- The minimum value of C_{MVS} remains unchanged for 10 images.

Finally, the image with the minimum value of C_{MVS} is selected to be labeled by the user. In our experiments, the value of S is always no more than 1% of the whole database when either of the conditions is satisfied, therefore, the processing time for evaluating the criterion value is greatly reduced.

To sum up, in each round of relevance feedback, the mean version space active learning method performs the following operations:

1. Construct a candidate set using one of the two schemes, and select the image with the smallest criterion value in the set to be labeled by the user;
2. Update the SVM classifier using the newly obtained labeled example.

4. EXPERIMENTAL RESULTS

4.1 Parameters and Operation Settings

To test the performance of the proposed mean version space active learning method, we perform systematic experiments on a general-purpose image database consisting of 5,000 Corel images. The database is made up of 50 categories, such as beach, bird, mountain, jewelry, sunset, etc. Each of the categories contains 100 images of essentially the same content, which serve as the groundtruth. In our experiments, we use each image in the whole database as a query, and average the results over the 5,000 queries. Unless otherwise stated, the precision vs. rounds of feedback curve is used to evaluate the performance of various methods.

Before the retrieval process, we need to construct a feature vector to represent each image. Feature selection is a large open problem and might have a great impact on the results. In our current implementation, the feature vector is simply made up of color histogram [14] and wavelet feature [19] since we focus on the relative performance comparison. Color histogram is obtained by quantizing the HSV color space into 64 bins. To calculate the wavelet feature, we first perform 3-level Daubechies wavelet transform to the image, and then calculate the first and second order moments of the coefficients in High/High, High/Low, and Low/High bands at each level, thus obtain an 18-dimensional feature. We will leave the problem of selecting the optimal feature combination to future work.

4.2 Evaluation of the Criterion

To compare the proposed mean version space criterion with that of SVM$_{\text{active}}$ and the most relevant strategy, we design the following experiment: given the query image, in the first round of relevance feedback, the system selects 10 images and asks the user for their labels; while in subsequent rounds of relevance feedback, only one image will be selected. Note that in the first round of relevance feedback, since there is only one positive example (the query image) and no negative one, no classifier is constructed and the system always presents the most relevant images to the user. We calculate the average P20[1] of mean version space (MVS), SVM$_{\text{active}}$, and the most relevant strategy (MR) after each round of relevance feedback, and compare their results in Figure 2 and Figure 3.

In our experiments, for fair comparison, we adopt the polynomial kernel with $p=1$ in SVM[2] (as in [17]), and construct both unbiased (Figure 2) and biased SVM (Figure 3) to obtain the separating hyperplane. The conclusion is the same: our mean version space criterion outperforms both SVM$_{\text{active}}$ and the most relevant strategy, which is consistent with the theoretical analysis. Take Figure 3 as an example, where we use biased SVM to learn the query concept. After the fifth round of relevance feedback, P20 using MVS is 0.268, using SVM$_{\text{active}}$ 0.244, and using MR 0.251.

It is interesting to note that in Figure 3, after the second round of relevance feedback, both SVM$_{\text{active}}$ and MR bring significant degradation to the performance although more labeled examples are available. However, when we use MVS to select images, no significant degradation in performance is observed; after the second round of relevance feedback, P20 is consistently improved.

[1] The reason for using P20 to compare the performance of different methods is that with many search engines, the first 20 images can be displayed in one page.

[2] ***Lemma 1*** requires that the feature space F is finite dimensional. Polynomial kernel satisfies this requirement, while RBF kernel does not [17].

4.3 The Candidate Set

The processing time of both SVM$_{\text{active}}$ and MR is very short, since they only need to construct one classifier in each round of relevance feedback. However, in the original form of MVS, we need to construct two classifiers for each unlabeled example, which results in a very long processing time.

In this subsection, we will evaluate MVS with the acceleration schemes when the next example to be labeled by the user is chosen from a small candidate set. As in the previous subsection, the system returns ten images for the user to label in the first round of relevance feedback, and returns only one image in subsequent rounds. Again, we use average P20 to compare the two simple schemes with the original method, and present their results in Figure 4 and Figure 5. (MVS denotes the original method, MVS(CB) denotes the scheme that selects the images closest to the boundary, while MVS(MR) denotes the scheme that selects the most relevant images)

Obviously, P50 of MVS(MR) is nearly identical with that of MVS, while P20 of MVS(CB) is inferior to both of them. The reason might be explained as follows: positive examples are often surrounded by negative ones, thus the most relevant images always lie in the neighborhood of those positive examples. On the other hand, with MVS(CB), we cannot guarantee that any selected example is in the neighborhood of labeled ones. Based on experimental results, we will use MVS(MR) as the acceleration scheme in subsequent experiments.

The advantage of using a small candidate set is a great reduction in processing time. In Table 1, we compare the average processing time of SVM$_{\text{active}}$, MR, MVS and MVS(MR) in the second round of relevance feedback **(**Pentium 4 1.80GHz, 512M RAM**)**. Obviously, when we use the candidate set, the processing time reduces to the same magnitude as that of SVM$_{\text{active}}$ and MR. Furthermore, the original method which evaluates the criterion for all the unlabeled images does not scale well; however, in the accelerated version, the processing time is mainly determined by S, which has no direct relationship with the size of the database.

Table 1. Comparison of processing time

	SVM$_{\text{active}}$	**MR**	**MVS**	**MVS (MR)**
Seconds	0.031	0.031	2.264	0.055

4.4 Feedback with Multiple Images

Although our analysis is for the case where only one example is labeled by the user in each round of relevance feedback, the mean version space active learning method also generalizes well when multiple images are labeled in each round. Figure 6 and Figure 7 compare the precision vs. scope curve of various methods after the fourth and fifth rounds of relevance feedback respectively, with ten images labeled by the user in each round. Note that the trained classifier is biased SVM with polynomial kernel ($p = 5$).

From the comparison results we can see that when multiple images are fed back in each round, the performance of SVM$_{\text{active}}$ is much worse than that of MR, while MVS consistently outperforms both MR and SVM$_{\text{active}}$. For example, after the fourth round of relevance feedback (Figure 6), P20 using MVS is 0.464, using MR 0.448, and using SVM$_{\text{active}}$ 0.364; while after the fifth round of relevance feedback (Figure 7), P20 using MVS is 0.537, using MR 0.515, and using SVM$_{\text{active}}$ 0.420.

We have also performed experiments with the RBF kernel. Again MVS outperforms both SVM$_{\text{active}}$ and MR no matter what value σ takes.

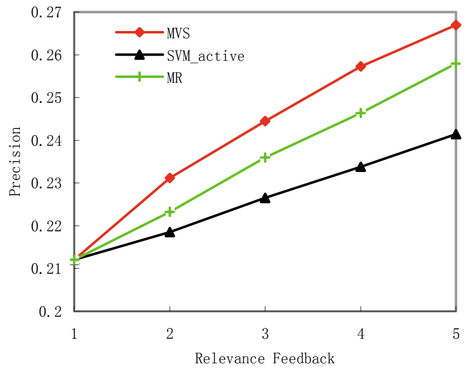


Figure 2. Performance comparison (unbiased SVM).

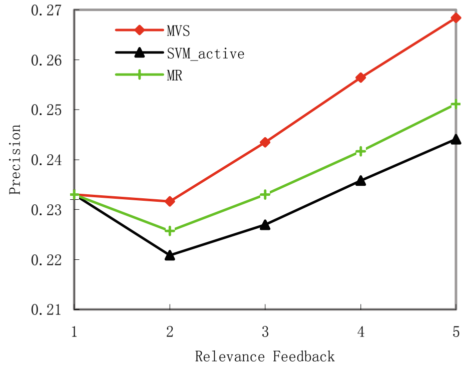


Figure 3. Performance comparison (biased SVM).

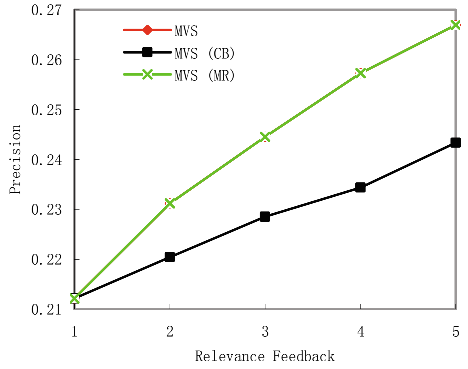


Figure 4. Performance comparison (unbiased SVM, the curve of MVS is overlapped by that of MVS(MR)).

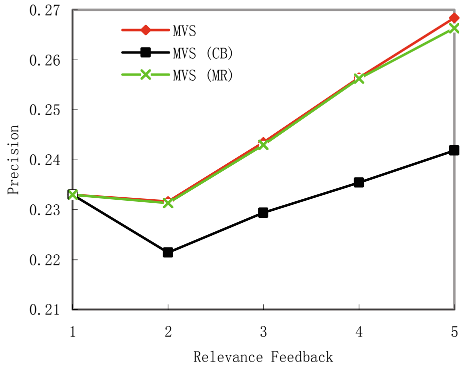


Figure 5. Performance comparison (biased SVM, the curve of MVS is overlapped by that of MVS(MR)).

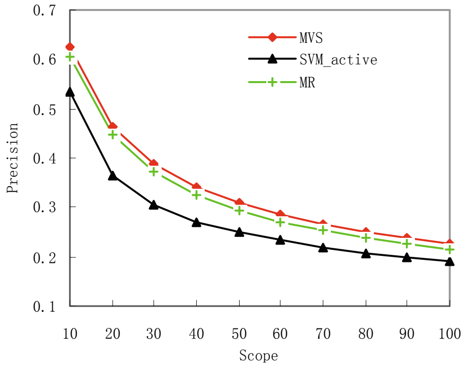


Figure 6. Performance comparison after the fourth round of relevance feedback.

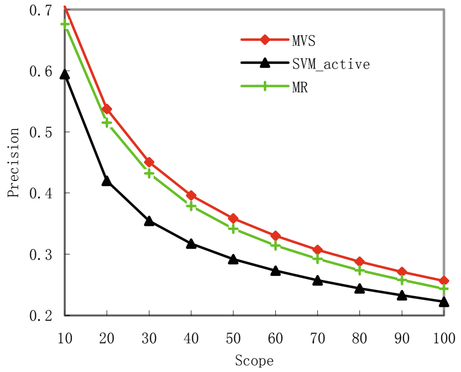


Figure 7. Performance comparison after the fifth round of relevance feedback.

5. CONCLUSION

In this paper, we have proposed a novel active learning method named mean version space, which is tailored for each specific learning task and can maximally shrink the version space. Our method takes both the size of the version space and the posterior probabilities into consideration, while $\text{SVM}_{\text{active}}$ and the most relevant strategy are only based on one of them. Our criterion is justified with both unbiased and biased SVM classifiers, and can be fitted in a general active learning framework. Furthermore, to reduce the processing time, we design two schemes to construct a candidate set in each round of relevance feedback and select images from this set. This operation is based on the observation that the location of the selected unlabeled example will affect the change in the separating hyperplane and also the accuracy of the posterior probabilities. We have evaluated the effectiveness of the mean version space method from various aspects by means of systematic experiments, which validate the advantage of our method over existing ones.

6. ACKNOWLEDGMENTS

This work was supported by National High Technology Research and Development Program of China (863 Program) under contract No.2001AA114190.

7. REFERENCE

[1] Cohn, D.A., Ghahramani, Z., and Jordan, M.I. Active learning with statistical models. *Journal of Artificial Intelligence Research*, vol. 4, pp. 129-145, 1996.

[2] Huang, J., et al. Image indexing using color correlograms. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 762-768, 1997.

[3] Liu, F., and Picard, R.W. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, 1996.

[4] Manjunath, B.S., and Ma, W.Y. Texture features for browsing and retrieval of image data. *IEEE Trans. on Pattern Anaysis and Machine Intelligence*, vol. 18, pp. 837-842, 1996.

[5] Mitchell, T. Generalization as search. *Artificial Intelligence*, vol. 28, pp. 203-226, 1982.

[6] Nguyen, H.T., Smeulders, A. Active learning using pre-clustering. *Proc. 21th Int. Conf. on Machine Learning*, 2004.

[7] Pass, G. Comparing images using color coherence vectors. *Proc. 4th ACM Int. Conf. on Multimedia*, pp. 65-73, 1997.

[8] Platt, J.C. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, MIT Press, 1999.

[9] Porkaew, K., and Chakrabarti, K. Query refinement for multimedia similarity retrieval in MARS. *Proc. 7th ACM Int. Conf. on Multimedia*, pp. 235-238, 1999.

[10] Rui, Y., et al. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE trans. Circuits and Systems for Video Technology*, 1998.

[11] Schimid, C. A structured probabilistic model for recognition. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 490, 1999.

[12] Schmid, C., and Mohr, R. Local grayvalue invariants for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 530-535, 1997.

[13] Schohn, G., and Cohn, D. Less is more: active learning with support vector machines. *Proc. 17th Int. Conf. on Machine Learning*, pp. 839-846, 2000.

[14] Swain, M., and Ballard, D. Color indexing. *Int. Journal of Computer Vision*, 7(1): 11-32, 1991.

[15] Tieu, K., and Viola, P. Boosting image retrieval. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 228-235, 2000.

[16] Tong, S., and Chang, E. Support vector machine active learning for image retrieval. *Proc. 9th ACM Int. Conf. on Multimedia, 2001.*

[17] Tong, S., and Koller, Daphne. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, vol. 2, pp. 45-66, 2001.

[18] Vapnik, V. *The Nature of Statistical Learning*. Springer, 1995.

[19] Wang, J.Z., Wiederhold, G., Firschein, O., and Sha, X.W. Content-based image indexing and searching using Daubechies' wavelets. *Int. Journal of Digital Libraries*, vol. 1, no. 4, pp. 311-328, 1998.

[20] Wang, L., Chan, K.L., and Zhang, Z. Bootstrapping SVM active learning by incorporating unlabelled images for image retrieval. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 629-634, 2003.

[21] Wu, Y., Tian, Q., and Huang, T. Discriminant-EM algorithm with application to image retrieval. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 155-162, 2000.

[22] Zhang, L., Lin, F., and Zhang, B. Support vector machine learning for image retrieval. *Proc. IEEE Int. Conf. on Image Processing*, vol. 2, pp. 721-724, 2001.

[23] Zhou, X.S., Rui, Y., and Huang, T. Water-Filling: a novel way for image structural feature extraction. *Proc. IEEE Int. Conf. on Image Processing*, vol. 2, pp. 570-574, 1999.