Dynamic Feature Space Selection in Relevance Feedback Using Support Vector Machines*

Fang Qian[1], Mingjing Li[2], Lei Zhang[2], Hongjiang Zhang[2], and Bo Zhang[3]

[1] State Key Lab of Intelligent Technology and Systems
Tsinghua University, Beijing 100084, China

[2] Microsoft Research Asia
49 Zhichun Road, Beijing 100080, China

[3] State Key Lab of Intelligent Technology and Systems
Tsinghua University, Beijing 100084, China

Abstract. The selection of relevant features plays a critical role in relevance feedback for content-based image retrieval. In this paper, we propose an approach for dynamically selecting the most relevant feature space in relevance feedback. During the feedback process, an SVM classifier is constructed in each feature space, and its generalization error is estimated. The feature space with the smallest generalization error is chosen for the next round of retrieval. Several kinds of estimators are discussed. We demonstrate experimentally that the prediction of the generalization error of SVM classifier is effective in relevant feature space selection for content-based image retrieval.

1 Introduction

Relevance Feedback (RF) has been regarded as an efficient technique to reduce the semantic gap via human-computer interaction in content-based image retrieval (CBIR). The pioneering works on relevance feedback focus on query point movement and similarity measure refinement [7, 13, 12]. Those approaches are based on the Euclidean distance or its variations, thus can be grouped into geometric methods. Lately proposed statistical approaches can be divided into probability-based and classification-based methods. Probability-based approaches are based on the MAP (*Maximum A Posterior*) criteria, with Cox [5],Vasconcelos [21] and Su [15] as the representatives. During the process of classification-based relevance feedback, a classifier is dynamically trained on the user-labeled positive and negative images, which then partitions the images in the database into two classes, either relevant or irrelevant. Once relevance feedback is treated as a learning problem, many classifiers can be applied, such as support vector machines (SVM) [6, 25], neural networks [26], Adaboost [16] and so on. Meanwhile the two fundamental problems in machine

* This work was performed at Microsoft Research Asia.

Qian F., Li M., Zhang L., Zhang H. and Zhang B. (2004).
Dynamic Feature Space Selection in Relevance Feedback Using Support Vector Machines.
In *Proceedings of the 4th International Workshop on Pattern Recognition in Information Systems*, pages 186-195
DOI: 10.5220/0002674001860195

learning, i.e., the selection of relevant features and training samples [1], also become important factors in the performance of relevance feedback. Most research works on classification-based relevance feedback can be grouped along these two branches.

For relevance feedback, the training samples are usually insufficient. The insufficiency is two-fold: (1) the number of training samples is too small compared with the overall size of image database; (2) the training samples are the nearest ones to the query, while images relevant in semantics might be spread out in the entire feature space. Therefore, the training samples are not necessarily representative, and hence relevant images far away from the query are very unlikely to be retrieved in the following iterations of relevance feedback. To make up the small sample problem, some works combine unlabeled data with labeled ones for training. For example, Wu [23] proposed Discriminant-EM algorithm within the transductive learning framework. While the results are promising, the computation may be a concern for large datasets. For the unrepresentative training samples problems, i.e., the labeled most positive images are not most informative, active learning may be an effective method. It tries to get more informative samples from users by actively selecting samples and requiring users to label. Tong and Chang [17] proposed the SVM active learning algorithm for relevance feedback in image retrieval. The points near the SVM boundary are used to approximate the most informative points and they are provided to users for labeling instead of the most positive images. In this way, the algorithm grasps the user's query concept accurately and quickly. Chang proposed the Maximizing Expected Generalization Algorithm (MEGA) [3]. To ensure that target concepts can be learned with a small number of samples, MEGA employs an intelligent sampling scheme that can gather maximum information for learning the user's concepts. MEGA judiciously selects samples at each iteration and uses positive samples to learn the target concept. At the same time, negative samples are used to shrink the candidate sampling space. In our previous work [11], we also proposed a relevance feedback approach to make the limited training samples more representative. The basic idea is to let the labeling of training samples and the training of the classifier be conducted in two complementary feature spaces respectively. In this way, the diversity of training samples is increased and thus the retrieval performance is improved.

For the feature selection problem, relevance feedback also has its own special issues. The dimensions of features used in CBIR are usually high, and several kinds of features are often combined to achieve better performance. Thus selecting the most relevant features is necessary to meet the real-time requirement. However, selecting the most relevant features for a specific query image is always a difficult problem in CBIR. Tieu and Viola [16] used more than 45,000 "highly selective features", and a boosting technique to learn a classification function in this feature space. Weak two-class classifiers are formulated based on Gaussian assumption for both the positive and negative examples along each feature component, independently. The strong classifier is a weighted sum of the weak classifiers as in AdaBoost. The disadvantage of their method is that too many features need to be conducted and stored, thus it is not practical.

In this paper, we propose an approach for dynamically selecting the most relevant feature space in relevance feedback. We employ the widely used classifier, support vector machines. The idea of selecting feature space is intuitive. During the feedback process, an SVM classifier is constructed in each feature space. The generalization error of each SVM is estimated, and used as a measure for feature space selection. The feature space with the smallest generalization error is chosen for the next round of retrieval. Experimental results demonstrate that this method improves the retrieval efficiency significantly with a small sacrifice of the retrieval effectiveness. A byproduct of our contribution is the comparison of three generalization error bounds.

The rest of the paper is organized as follows. The support vector machines and the estimators of generalization error are briefly introduced in Section 2 and Section 3 respectively. The proposed relevance feedback algorithm based on dynamically selecting feature spaces is described in Section 4. Experimental results are presented in Section 5 and the concluding remarks are given in Section 6 finally.

2 Support Vector Machines

Support vector machines are based on the Structural Risk Minimization (SRM) principle from statistical learning theory [18, 19]. Given the training data $(\mathbf{x}_i, y_i)$, i=1,.., n., $\mathbf{x}_i \in R^d$, $y_i \in \{-1,+1\}$, SVM maps the input vectors $\mathbf{x}$ into a high-dimensional feature space $\mathbf{H}$ through some mapping function ϕ : $\mathbf{R}^d \mapsto \mathbf{H}$, and constructs an optimal separating hyperplane in this space [2]. The mapping $\phi(\cdot)$ is implemented by a kernel function $K(\cdot,\cdot)$ which defines an inner product in $\mathbf{H}$. The decision function of an SVM is as following:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i^* y_i K(\mathbf{x}_i \cdot \mathbf{x}) + b \tag{1}$$

The $\boldsymbol{\alpha}^*$ is the optimal solution of the following optimization problem:

$$\text{Maximize: } W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{2}$$

$$\text{Subject to: } \sum_{i=1}^{n} y_i \alpha_i = 0 \tag{3}$$

$$\forall_{i=1}^{n} : 0 \le \alpha_i \le C \tag{4}$$

The factor C in (4) is a parameter that allows trading-off between training errors and model complexity.

SVM has been successfully applied to relevance feedback in image retrieval [6, 25]. An SVM captures the query concept by separating the relevant images from the irrelevant images with a hyperplane in a projected space, usually a very high-dimensional one. The projected points on one side of the hyperplane are considered relevant to the query concept and the rest irrelevant. Once the classifier is trained, SVM returns *k* images farthest from the hyperplane on the query concept side as top *k* most relevant images.

3 The Generalization Performance of SVM

For classification based relevance feedback, the retrieval performance depends on the generalization error of the classifier directly. The key idea of our approach is to dynamically select a feature space for retrieval in each round of feedback, in which the estimated generalization error is minimal. To get an estimate of the generalization error, we study several bounds of the expected error probability of SVMs.

3.1 Radius-Margin Bound

If *n* training samples belonging to a sphere of radius *R* are separable with the corresponding margin *M*, then the expectation of the error probability has the bound [20]

$$EP_{err} \le \frac{1}{n} E\left\{ \frac{R^2}{M^2} \right\} \tag{5}$$

where the expectation is taken over all training sets of size *n*.

3.2 The Number of Support Vectors

Vapnik [18] gives an alternative bound on the actual risk of SVMs:

$$EP_{err} \le E\left\{ \frac{N_{SV}}{n} \right\} \tag{6}$$

where N_{SV} denotes the number of support vectors.

3.3 ξα – estimator

This estimator is proposed by Joachims for text classification [8, 9]. $\xi\alpha$ – estimators are based on the idea of leave-one-out (LOO) estimation, but overcome the computation disadvantage of LOO. The estimator is named by the two arguments: $\boldsymbol{\xi}$ and $\boldsymbol{\alpha}$. $\boldsymbol{\xi}$ is the vector of training losses at the solution of the primal SVM training problem. If a training example lies on the "wrong" side of the hyperplane, the corresponding ξ_i is greater than or equal to 1. The training losses ξ_i can be computed as $\xi_i = \max(1 - y_i(\mathbf{w} \cdot \mathbf{x}_i + b), 0)$. $\boldsymbol{\alpha}$ is the solution of the dual SVM problem. Both $\boldsymbol{\xi}$ and $\boldsymbol{\alpha}$ are available after training the SVM at no extra cost.

For stable soft-margin SVMs, the $\xi\alpha$ – estimator of the error rate is defined as:

$$EP_{err} \le \frac{d}{n} \text{ with } d = \left|\left\{ i : \left(2\alpha_i R_\Delta^2 + \xi_i\right) \ge 1 \right\}\right| \quad (7)$$

where R_Δ^2 is an upper bound on $K(\mathbf{x},\mathbf{x}) - K(\mathbf{x},\mathbf{x}')$ for all $\mathbf{x}$ and $\mathbf{x}'$, i.e. $R_\Delta^2 \ge \max_{\mathbf{x},\mathbf{x}'}\left(K(\mathbf{x},\mathbf{x}) - K(\mathbf{x},\mathbf{x}')\right)$.

The key idea to the $\xi\alpha$ – estimator is a connection between the training examples for which the inequality $2\alpha_i R_\Delta^2 + \xi_i \ge 1$ holds and those training examples that can produce an error in leave-one-out testing. It can be proved that *d* is an upper bound on the number of leave-one-out errors.

4 Dynamic Feature Space Selection

The central idea of our algorithm is based on the relationship between the relevant degree of a feature space and the generalization ability of the classifier trained in this space. If the classifier in one feature space achieves smaller generalization error, we can say that this feature space is more relevant to the current classification problem.

Given M features: $\Phi_1,\cdots\Phi_M$, denote the combined feature as Φ_0=[$\Phi_1,\cdots\Phi_M$]. Note that all kinds of features are normalized into the same range. The detailed algorithm can be described as follows:

Step 1: The system presents initial results in Φ_0 space and the user labels top *k* images;

Step 2: In each feature space $\Phi_1,\cdots\Phi_M$, an SVM classifier is constructed separately;

Step 3: Estimating the generalization error of each SVM respectively, denote as $T_1,\cdots,T_M$;

Step 4: Choosing the feature space in which the generalization error is minimized;

$$J = \arg\min_i T_i,\ i = 1,\cdots,M \quad (8)$$

Step 5: In the selected space Φ_J, SVM classifies all images in the database and returns top *k* most relevant images for user labeling;

Step 6: Repeat from Step 2 to Step 5 until the user stops relevance feedback.

Estimates of the generalization error of SVMs have been employed for feature selection [22]. In their work, feature selection problem can be formulated as a preprocessing of the data $\mathbf{x} \mapsto (\mathbf{x} * \sigma),\ \sigma \in \{0,1\}^n$. Therefore the feature selection problem is transferred to calculate suitable parameters σ. This is done by minimizing some estimates of the generalization error of SVMs using a gradient descent algorithm. Comparing with their method, our approach is simple and direct, well meeting the real-time requirement in relevance feedback.

5 Experimental Results

5.1 Experimental Design

The image database we used consists of 10,000 images from the Corel dataset. It is a large and heterogeneous image set. Images from the same category as that of the query are used as the ground truth. Relevance feedback is conducted automatically. In the first iteration of feedback, top 30 images are checked and labeled as either positive or negative examples. In the following iterations, the labeled positive images are ranked at the beginning, while the negative images are ranked at the end. All of the positive and negative images in each round are accumulatively used.

The most commonly used kernel function, RBF kernel, is selected for SVMs: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-g\|\mathbf{x}_i - \mathbf{x}_j\|^2)$. The kernel parameter *g* needs to be pre-defined. We will choose the best *g* experimentally for each feature space. Note that the control factor C is set to 100 for all feature spaces. Since $K(x,x)=1$ in RBF kernel function, R_Δ^2 in $\xi\alpha-$ estimator is set to 1.

5.2 Results on "Bead" Category

To demonstrate the effectiveness of dynamically selecting feature space more clearly, we take the "*bead*" category as an example. "Bead" category contains images of beads with different colors and shapes, as shown in Figure 1. However, these images are similar in texture. That is to say, texture feature is relevant while color feature is irrelevant for this category.

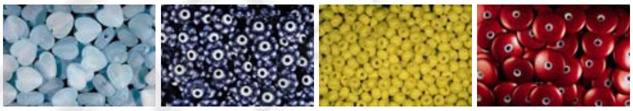

Fig. 1. Random images from bead category

Two features are used in this experiment: Color Histogram (CH) [24] and Gabor-based texture feature (Gabor) [10]. Since the most relevant feature for "bead" category is texture feature, the relevance feedback algorithm is expected to select the Gabor texture feature dynamically to perform the retrieval. Table 1 listed the selected kernel parameters for each feature space by experiments, where CH-Gabor represents the feature concatenated by CH and Gabor.

Table 1. The parameters of SVM in each feature space

	CH	Gabor	CH-Gabor
g	2.0	0.5	1.0

We compared the retrieval performance of three methods on "*bead*" category. The "alternating model" refers to the method that alternates the feedback and retrieval operation between CH color feature and Gabor texture feature [11]. The "flat model" means that CH and Gabor features are merged into a new feature CH-Gabor for retrieval and feedback. The "dynamic model" means that at each round of feedback, the retrieval operation is switched to the most relevant feature space according to the bound of generalization error. The bound used in Figure 2 is $\xi\alpha$ – estimator. For all three methods, SVM based classification is used as feedback approach. CH-Gabor feature space is used at the first retrieval to give the three methods a same start point.

Figure 2 shows the average retrieval performance of three methods on 100 bead images. From Figure 2, we can observe that:

- The performance curve of the alternating model waves with the relevance feedback going on. The performance drops when the retrieval operation is changed into CH color feature space; meanwhile, the performance is improved largely when the retrieval is performed in Gabor texture feature space. The reason is that for the current query concept, CH feature is weakly relevant even irrelevant while Gabor texture feature is strongly relevant.
- The performance of the dynamic model and the flat model has a consistent increase. Both methods improve the initial retrieval performance significantly after several rounds of relevance feedback. The dynamic model lags behind the flat model because it cuts the weakly relevant features for speed up.

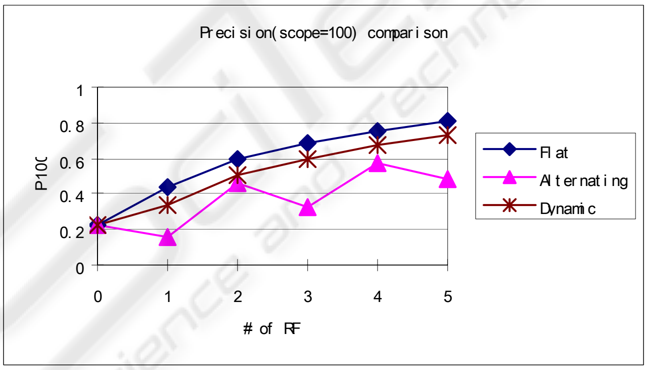

Fig. 2. Performance comparison on "*bead*" category

As to the computation efficiency, we compared the average retrieval time for each query image during the five rounds of relevance feedback. The cost time of the flat model is 8.80 seconds, the alternating model 3.65 seconds and the dynamic model 3.96 seconds. The above results demonstrate that the dynamic model improves the retrieval efficiency significantly with only a small sacrifice of the retrieval effectiveness.

In order to check whether the dynamic model can select the most relevant feature space correctly during the relevance feedback, Table 2 compares the selected times of two feature spaces at each round of RF. For total 100 queries, at the first round of RF, only 60 ones are judged to be suitable in Gabor texture feature space. However, as the feedback goes on, the Gabor texture feature space is selected more frequently. At the fifth round of RF, only 3 queries are retrieved in CH color space. This demonstrates that in the feedback process, the dynamic model selects the relevant texture feature space correctly. The non-ideal performance at the first round of RF can be explained by the insufficient samples, which make the estimation inaccurate.

Table 2. The selected times of two feature spaces at each round of RF

	RF1	RF2	RF3	RF4	RF5
CH	40	16	11	5	3
Gabor	60	84	89	95	97

5.3 More General Results

The purpose of this experiment is to compare the effectiveness of three generalization bounds. In this experiment, the query set is expanded into the following ten categories: aquarelle, bead, building, dish, flag, horse, mountain, road sign, ski and sky, totally 100 images. Three kinds of low-level features are used: Color Histogram (CH) [24], Color Moments (CM) [14] and Wavelet based Texture feature (WT) [12]. The kernel parameters for each space are listed in Table 3. At each round of RF, the dynamic model selects one of them according to the three measures described in Section 4. The alternating model changes the space according to a fixed order: CM-WT-CH. Other orders (eg. CM-CH-WT) give similar results, so we do not present here.

Table 3. The parameters of SVM in each feature space

	CH	CM	WT
g	2.0	5.0	5.0

We can obtain the following observations from Figure 3:

- All three bounds of generalization error are useful to the feature space selection. The dynamically selecting feature space methods are better than the fixed alternating feature space methods.
- Among the three kinds of bound, the $\xi\alpha-$estimator performs best, the number of support vectors performs worst, and the radius-margin bound performs between them.

Regarding the computation efficiency, both the number of SVs and the $\xi\alpha-$estimator can be obtained at no extra cost immediately after training SVM. However, the radius-margin bound needs a little more time to solve the optimization problem for radius *R*. Therefore, we recommend $\xi\alpha-$estimator considering both its effectiveness and efficiency.

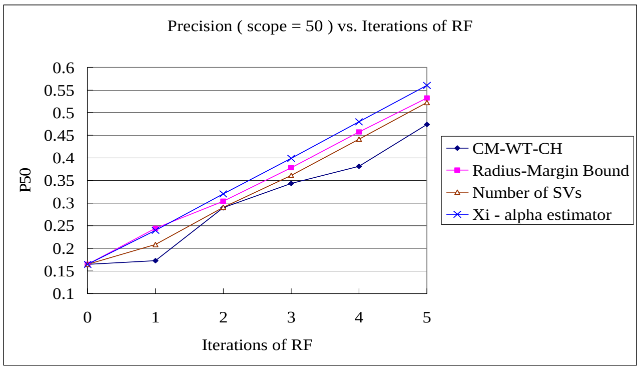

Fig. 3. Performance comparison results over general categories

6 Conclusion

We have proposed an approach for dynamically selecting feature space in relevance feedback. During the feedback process, a support vector machine is constructed in each feature space. The generalization error of each SVM is estimated. Among all feature spaces, the one with the smallest generalization error is chosen for the next round of retrieval. The experimental results demonstrated the proposed algorithm could improve the retrieval efficiency significantly with only a small sacrifice of the retrieval effectiveness. We also studied three estimates of the generalization error: the radius-margin bound, the number of support vectors and the $\xi\alpha-$estimator. The $\xi\alpha-$estimator is the best in terms of efficiency and effectiveness.

References

1. Blum, A. and Langley, P. (1997) Selection of relevant features and examples in machine learning. Artificial Intelligence, 97:245-271
2. Burges, C.J.C. (1998) A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):955-974
3. Chang, E. and Li, B. (2001) MEGA --- The Maximizing Expected Generalization Algorithm for learning complex query concepts, February. UCSB Technical Report, http://www-db.stanford.edu/~echang/mega-extended.pdf
4. Chapelle, O., Vapnik, V., Bousquet, O. and Mukherjee, S. (2001) Choosing multiple parameters for support vector machines. Machine Learning
5. Cox, I. J. , Minka, T. P., Papathomas, T.V. and Yianilos, P. N. (2000) The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments, IEEE Transactions on Image Processing – special issue on digital libraries
6. Hong, P., Tian, Q. and Huang, T. S. (2000) Incorporate support vector machines to content-based image retrieval with relevance feedback, IEEE Int'l conf. on Image Processing (ICIP'2000), Vancouver, Canada
7. Ishikawa, Y. and Subramanya, R. (1998) MindReader: Query databases through multiple examples, in Proc. of the 24th VLDB conference, (New York)

8. Joachims, T. (2000) Estimating the generalization performance of a SVM efficiently. Proceedings of the Seventeenth International Conference on Machine Learning. 2000. San Francisco: Morgan Kaufman
9. Klinkenberg, R. and Joachims, T. (2000) Detecting concept drift with support vector machines. In Proceedings of the Seventeenth International Conference on Machine Learning. 2000. San Francisco. Morgan Kaufmann
10. Manjunath, B.S., and Ma, W.Y., Texture features for browsing and retrieval of large image data, IEEE Transactions on Pattern Analysis and Machine Intelligence, 18, 837-842, 1996
11. Qian, F., Li, M., Ma, W., Lin, F. and Zhang, B. (2001) Alternating feature spaces in relevance feedback, 3rd International Workshop on Multimedia Information Retrieval, October 5, 2001, Ottawa, Canada
12. Rui Y. and Huang T. S. (2000) Optimizing learning in image retrieval. Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR), Hilton Head, SC
13. Rui, Y., Huang, T. S., Ortega, M. and Mehrotra, S. (1998) Relevance feedback: A power tool in interactive content-based image retrieval, IEEE Transaction on Circuits and Systems for Video Technology, Special Issue on Segmentation, Description, and Retrieval of Video Content, 8(5): 644-655
14. Stricker M. and Orengo, M., Similarity of color images. in Proc. SPIE Storage and Retrieval for Image and Video Databases. 1995
15. Su, Z., Zhang, H. J. and Ma, S. (2001) Relevant feedback using a Bayesian classifier in content-based image retrieval, SPIE Electronic Imaging 2001, San Jose, CA
16. Tieu, K. and Viola, P. (2000) Boosting image retrieval, Proc. IEEE Conf. Computer Vision and Pattern Recognition, Hilto Head Island, SC
17. Tong, S. and Chang, E. (2001) Support vector machine active learning for image retrieval, in Proc. ACM Multimedia 2001, Ottawa, Canada
18. Vapnik, V. (1995) The nature of statistical learning theory. Springer-Verlag, New York
19. Vapnik, V. (1998) Statistical learning theory. Chichester, GB: Wiley
20. Vapnik, V. and Chapelle, O. (2000) Bounds on error expectation for support vector machines. Neural Computation
21. Vasconcelos, N. and Lippman, A. (1999) Learning from user feedback in image retrieval systems, NIPS'99, Denver, Colorado
22. Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T. and Vapnik, V. (2001) Feature selection for SVMs. In Sara A Solla, Todd K Leen, and Klaus-Robert Muller, editors, Advances in Neural Information Processing Systems 13. MIT Press
23. Wu, Y., Tian, Q. and Huang, T. S. (2000) Discriminant-EM algorithm with application to image retrieval, in Proc. IEEE Conf. Computer Vision and Pattern Recognition, South Carolina
24. Zhang, L., Lin, F. and Zhang, B. (1999) A CBIR method based on color-spatial feature. IEEE Region 10 Annual International Conference 1999 (TENCON'99), Cheju, Korea. 1999:166-169
25. Zhang, L., Lin, F. and Zhang, B. (2001a) Support vector learning for image retrieval, IEEE International Conference on Image Processing (ICIP 2001). pp721-724. Thessaloniki, Greece
26. Zhang, L., Lin, F. and Zhang, B. (2001b) A neural network based self-learning algorithm of image retrieval, Chinese Journal of Software, 12(10): 1479-1485