INCORPORATE SUPPORT VECTOR MACHINES TO CONTENT-BASED IMAGE RETRIEVAL WITH RELEVANT FEEDBACK[†]

Pengyu Hong, Qi Tian, Thomas S. Huang
IFP Group, Beckman Institute
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
{hong, qitian, huang}@ifp.uiuc.edu

ABSTRACT

By using relevance feedback [6], *Content-Based Image Retrieval (CBIR) allows the user to retrieve images interactively. The user can select the most relevant images and provide a weight of preference for each relevant image. The high level concept borne by the user and perception subjectivity of the user can be captured by the system to some degree. This paper proposes an approach to utilize both positive and negative feedbacks for image retrieval. Support Vector Machines (SVM) is applied to classifying the positive and negative images. The SVM learning results are used to update the preference weights for the relevant images. This approach releases the user from manually providing preference weight for each positive example. Experimental results show that the proposed approach has improvement over the previous approach* [5] *that uses positive examples only.*

1. Introduction

Digital image retrieval systems allow sophisticated querying and searching by image content. Since 1990's, Content-Based Image Retrieval (CBIR) has attracted great research attention [3][4][7]. Early research focused on finding the "best" representation for image features. The similarity between two images is calculated by summing the distances of individual low-level features with fixed weights. In this context, high-level concepts and user's perception subjectivity cannot be well modeled. Recent approaches introduce human-computer interaction into CBIR [2][6][9]. The interactive mechanism [6] allows the user to submit a coarse initial query and continuously refine his information need *via* relevance feedback. The weights of the low-level visual features are updated based on the feedback. This approach greatly reduces the labor required to compose a query and captures the user's information need more precisely.

However, [5][6] only uses the positive examples as feedback. The information implied by the negative examples is neglected. Moreover, MARS [5] requires the user to provide preference weights of the relevant images, which sometimes is difficult for the user to give a clear choice. An example of using both the positive and negative examples, which are chosen by the user, for image retrieval can be found in FourEyes [8]. The system looks at all the local models and determines which model or combination of models best covers the positive examples, while satisfying the constraints implied by the negative examples.

In this paper, we propose to apply Support Vector Machine to two classes (positive and negative examples) learning. The learning results are further used to help automatically decide preference weights for the positive images. The rest of paper is organized as follows. Beginning with the discussion of relevance feedback technique in section 2, we briefly describe Support Vector Machines (SVM) in section 3. The application of SVM to CBIR is explained in section 4. The proposed methods are tested and the experimental results are given in section 5. Finally, section 6 is the summery.

2. Relevance Feedback

Relevance feedback is a technique that takes advantage human-computer interaction to refine high level queries represented by low level features. It is used in traditional document retrieval [1] for automatically adjusting an existing query using information fed back from the user. In the application of image retrieval [6], the user selects relevant images from previous retrieved results and provides a preference weight for each relevant image. The weights for the low-level feature, i.e., color and texture, etc., are dynamically updated based on the user's feedback. The user is no longer required to specify a precise weight for each low-level feature at the query formulation stage. Based on user's feedback, the high

---

level concepts implied by the query weights are automatically refined.

During the process of relevant feedback, the similarity between the query (relevant images) and those in the database are calculated. The overall similarity between an image *I* in the database and the relevant images is calculated by:

$$S(I) = \sum_f w_f F_f(I) \qquad (1)$$

where $F_f(I)$ is the similarity of individual feature (*e.g.* color, texture, *etc*). *Mahalanobis* distance is used as the similarity measurement:

$$F_f(I) = (\vec{x}_f - \vec{q}_f)^T C_f^{-1} (\vec{x}_f - \vec{q}_f) \qquad (2)$$

where $\vec{x}_f$ is the *f*th feature vector of the image *I*, $\vec{q}_f$ is the *f*th feature vector of the query and C_i is the covariance matrix of the *f*th feature components of the query. $\vec{q}_f$ and C_i are decided by Eq. (3) and (4) respectively

$$\vec{q}_f = \sum_{k=1}^{N} v_k \vec{m}_{kf} \qquad (3)$$

$$C_i(m,n) = \frac{\sum_{k=1}^{N} v_k (\vec{m}_{kf} - \vec{q}_f)(\vec{m}_{kf} - \vec{q}_f)^T}{\sum_{k=1}^{N} v_k} \qquad (4)$$

where *N* is the number of relevant image (positive feedback), v_k is the preference weight for the *k*th relevant image (positive feedback), $\vec{m}_{kf}$ is the *f*th feature vector of the *k*th relevant image. C_i is set as an identity matrix if there is only one relevant image.

The Low-level feature weight w_f in Eq. (3) is updated by Eq. (5):

$$w_f = \frac{\sum_{k=1}^{N} v_k}{\sum_{k=1}^{N} v_k F_f(K)} \qquad (5)$$

The main idea behind Eq. (5) is that: the smaller the average feature distance over the relevant images, the better the feature represent the query concept. Hence, higher weight is given to the feature that has smaller average feature distance over the relevant images.

Currently, the system requires the user manually provide a preference weight v_k for each relevant image. The preference weights denote the degree of how much the user likes the images. Moreover, only positive examples are used. However, in some cases, there exist examples that are not desired by the user but closer to the query than some of the relevant images based on the

above calculation. Those examples will be retrieved, and their ranks may be even higher than some relevant images. Hence, it is important to use the information implied by the negative examples. Moreover, expressing the perception subjectivity *via* providing numerical preference weights is a difficult task for the users from time to time. The query may be harmed by inappropriate assignments of preference weights.

In this paper, we proposed to use SVM to perform non-linear classification on the positive and negative feedbacks. The learning results are utilized to automatically calculate the preference weights. In this way, we make the retrieval procedure simpler.

3. SUPPORT VECTOR MACHINES

Support Vector Machines (SVM) [11] is an approximate implementation of the *structural risk minimization* (SRM) principle. It creates a classifier with minimized *Vapnik-Chervonenkis* (VC) dimension. SVM minimizes an upper bound on the generalization error rate. The SVM can provide a good generalization performance on pattern classification problems without incorporating problem domain knowledge. Consider the problem of separating the set of training vectors belonging to two classes:

$$\{(\vec{x}_i, y_j)\}_{i=1}^{N}, \quad y_i = +1/-1 \qquad (6)$$

where $\vec{x}_i$ is an input pattern, and y_i is the label, +1 denotes positive example, -1 denotes the negative example. If those two classes are linearly separable, the hyperplane that does the separation can be easily calculated by:

$$\vec{w}^T \vec{x} + b = 0 \qquad (7)$$

where $\vec{x}$ is an input vector, $\vec{w}$ is a weight vector, and *b* is a bias. The goal of SVM is to find the parameters w_o and b_o for the optimal hyperplane to maximize the distance between the hyperplane and the closest data point:

$$\begin{aligned} w_o^T \vec{x}_i + b_o \ge 1 \qquad &\text{for } y_i = +1 \\ w_o^T \vec{x}_i + b_o < -1 \qquad &\text{for } y_i = -1 \end{aligned} \qquad (8)$$

A linear separable example in 2D is illustrated in Figure 1. If the two classes are non-linearly separable, the input vectors should be nonlinearly mapped to a high-dimensional feature space by an inner-product kernel function $K(\vec{x}, \vec{x}_i)$. Table 1 shows three typical kernel functions [10]. An optimal hyperplane is constructed for separating the data in the high-dimensional feature space. This hyperplane is optimal in the sense of being a maximal margin classifier with respect to the training data.
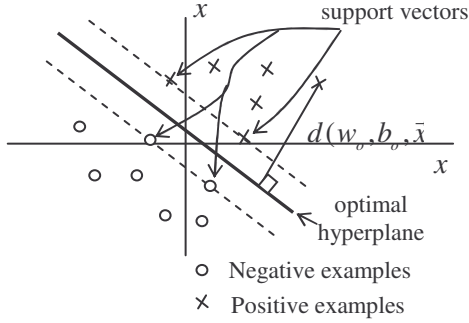
Figure 1. An example of the optimal hyperplane for linearly separable patterns

Kernel Function	Inner Product Kernel $K(\vec{x}, \vec{x}_i)$, $i = 1, 2, \ldots, N$
Polynomial	$(\vec{x}^T \vec{x}_i + 1)^P$, $p=1,2,\ldots$
Radial-basis	$\exp(-\dfrac{1}{2\sigma^2}\|\vec{x} - \vec{x}_i\|^2)$, σ is decided by the user
Sigmoid	$\tanh(\alpha_0 \vec{x}^T \vec{x} + \alpha_1)$, α_0 and α_1 are decided by the user

Table 1. Types of kernel functions

4. SVM IN CBIR

Usually, the problem to separate the negative examples from the positive examples turns out to be finding a nonlinear classifier. SVM can be used in this task, and it provides a good generalization performance at the same time. Given w_o and b_o, the distance of a point $\vec{x}$ from the optimal hyperplane is defined as

$$d(w_o, b_o, \vec{x}) = \frac{\left|w_o^T \vec{x} + b_o\right|}{\|w_o\|} \qquad (9)$$

The distance indicates how much an example belonging to one class is different from the other one. These motivate us to use SVM for automatically generating preference weights for relevant images. Intuitively, the farther the positive examples from the hyperplane, the more distinguishable they are from the negative examples. Thus, when we decide their preference weights, they should be assigned with larger weights. Currently, we simply set the relation between the preference weights and the distance as a linear relation in the numerical calculation. It can be easily extended to nonlinear relation. During the iterative query procedure, the positive and negative examples selected in the history are collected for learning at each query time.

5. EXPERIMENTAL RESULTS

We test the proposed approach on COREL dataset, which contains more than 17,000 images. The visual features extracted form images include color (color moments) and texture (wavelet moments). Polynomial kernel function with $p = 1$ is used for SVM learning. The weights are normalized to the range of 10-100.
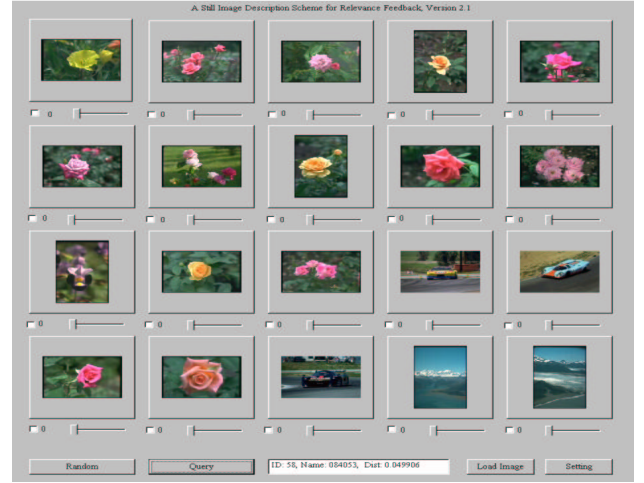

Figure 2. Retrieval results using positive feedbacks only (The images are sorted based on the similarity to the query. The ranks descend from left to right and from top to bottom.)
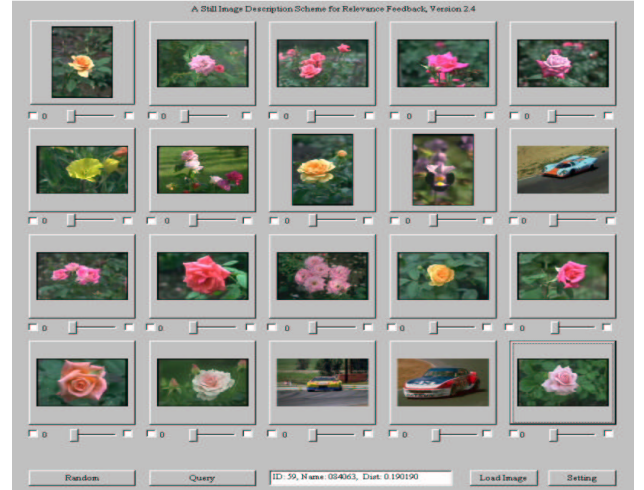

Figure 3. Retrieval results using the proposed approach. The organization of the image is same to that of Figure 2.

Figure 2 shows a retrieval result of flower images using positive feedbacks only. Figure 3 shows the results using the proposed approach. Both cases use same four positive feedbacks. Listed in the order of from top to bottom and from left to right, the positive examples are the 1st, 2nd, 3rd and 6th images in Fig. 2. Four negative examples are selected as negative feedback. One of them is the 19th

image in figure 2. The other three do not appear in the retrieval results. Two positive examples (the 3rd and 6th image in figure.2) and two negative examples are selected as supported vectors. The preference weights that are calculated based on the distances output by SVM learning.

The 1st image in Figure 2 has the largest distance to the hyperplane determined by the support vectors. Therefore, it is the most distinguishable from the negative examples. The largest weight 100 was assigned to it. The 3rd and 6th images of Figure 2 are selected as support vectors, smaller weights (10) are assigned to them. Compared to the results shown in Figure 2, more flowers are retrieved in the top 20 returned images shown in Figure 3.
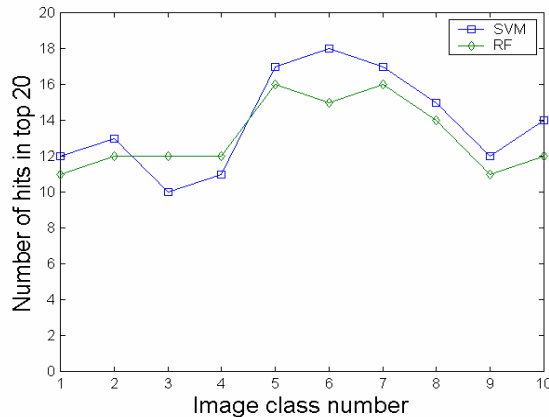

Figure 4. Compare the proposed approach and relevance feedback using positive examples only.

Tests are performed on ten class images (e.g. car, flower, airplane, *etc.*). Figure 4 compares the numbers of hits in top 20 returned images using our approach and those of using positive feedback only. The proposed approach shows improvement over using positive feedback alone in most cases. However, in some case, we got contrary results.

Enough number of positive and negative feedbacks is needed for reliable SVM learning. This is the major limit of this approach. Therefore, when the size of the query image set is small, we still use the positive feedbacks only. After more relevant images are returned, the proposed approach can be performed. Currently, we heuristically set the size threshold as at least 4 positive examples and 4 negative examples. Another issue is how to choose the kernel function. We leave this open for future investigation.

6. SUMMERY

This paper proposes to incorporate SVM into CBIR with relevant feedback. The information carried by positive and negative examples are explored by SVM learning. The learning results are used to automatically update preference weights for positive relevant images. This not only releases the users from providing accurate preference weight for each positive relevant image but also utilizes the negative information. Reasonable better results are obtained compared to that of positive feedbacks only.

8. REFERENCES

[1] C. Buckley and G. Salton, "Optimization of relevance feedback weights", in *Proc. Of SIGIR'95*.
[2] D. Daneels, et al. "Interactive Outlining: An Improved Approach Using Active Contours," in *Proc. SPIE Storage and Retrieval for Image and Video Databases*, 1993.
[3] C. Faloutsos, M. Flickner, et al., "Efficient and Effective Querying by Image Content," Tech. Rep. IBM Research Report, 1993.
[4] M. Flickner, H. Sawhney, et al., "Query by Image and Video Content: The QBIC System", *IEEE Computer*, 1995.
[5] Y. Rui, T. S. Huang, S. Mehrotra, "Content-Based Image Retrieval With Relevance Feedback in MARS", *Proc. Of IEEE Intl. Conf. on ICIP'97*, Santa Barbara, CA, 1997.
[6] Y. Rui, T. S. Huang, et al., "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval", *IEEE Trans. on Circuits and Video Tech.*, Special Issue on Segmentation Description, and Retrieval of Video Content, 8(5), Sep. 1998.
[7] A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Content-Based Manipulation of Image Databases", *Intl. Journal of Computer Vision*, 1996.
[8] R. Picard, "A Society of Models for Video and Image Libraries", IBM Systems Journal, vol. 35, no. 3&4, pp. 292-312, 1996.
[9] J.R. Smith and S.F. Chang, "An Image and Video Search Engine for the World-Wide Web," in *Proc. SPIE Storage and Retrieval for Image and Video Databases*, 1997.
[10] M. O. Stitson, J. A. E. Weston, et al., " Theory of Support Vector Machine", *Technical report, CSD-TR-96-17*, Univ. of London.
[11] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.