Using Bayesian Classifier in Relevant Feedback of Image Retrieval

Zhong Su[1], Hongjiang Zhang[2], Shaoping Ma[3]
[1,3]*Department of Computing Science, Tsinghua University, Beijing 100084, China*
[2]*Microsoft Research China 5F, Beijing Sigma Center Beijing 100080, China*
suzhong_bj@hotmail.com[1], *hjzhang@microsoft.com*[2], *msp@ tsinghua.edu.cn*[3]

Abstract

Relevance feedback is a powerful technique in content-based image retrieval (CBIR) and has been an active research area for the past few years. In this paper, we propose a new relevance feedback approach based on Bayesian classifier and it treats positive and negative feedback examples with different strategies. For positive examples, a Bayesian classifier is used to determine the distribution of the query space. A 'dibbling' process is applied to penalize images that are near the negative examples in the query and retrieval refinement process. The proposed algorithm also has the progressive learning capability that utilize past feedback information to help the current query. Experimental results show that our algorithm is effectiveness.

1. Introduction

While there are many research efforts in addressing content-based imaged retrieval (CBIR) [1][5][8], the performance of CBIR methods are still limited. One of the problems that pose performance limitations to CBIR is the disparity between semantic concepts and low-level image features. The mapping between them is still impractical with today computer vision and AI techniques. To improve this situation, more research efforts have been shifted to relevance feedback techniques recently.

Relevance feedback is a supervised learning technique used to improve the performance of information retrieval systems. Most of the previous relevance feedback researches can be classified into two classes: query point movement[2][12] and re-weighting[4]. The query point movement method essentially tries to improve the estimate of the "ideal query point" by moving it towards positive examples and away from negative ones. The frequently used technique to iteratively improve this estimation is the Rocchio's formula [7]. Experiments show that the retrieval performance can be improved considerably with such relevance feedback approaches [1][9][10].

The key idea behind the re-weighting method is very simple and intuitive. The MARS[7] system implemented a refinement to re-weighting methods, called standard deviation method. Each image can be viewed as a vector in a N dimensional feature space; and if the variance of positive examples is high along a principle axis j, then we can deduce that the values on this axis is not very relevant to the input query, thus, assigned a low weight w_j. Therefore, the inverse of the standard deviation of the j^{th} feature values in the feature matrix is used as the to update the weight w_j.

Recently, more computationally robust methods that perform global optimization have been proposed. The MindReader[4] retrieval system designed by Ishikawa et al. formulates a minimization problem on the parameter estimating process. Unlike traditional retrieval systems whose distance function can be represented by ellipses aligned with the coordinate axis, the MindReader system proposed a distance function that is not necessarily aligned with the coordinate axis. Therefore, it allows for correlations between attributes in addition to different weights on each component. A further improvement of this approach was proposed in [8]. This work not only formulates the relevance feedback as an optimization problem but also proposed a multi-level feature model.

In this paper we present a new relevance feedback algorithm based on Bayesian classifier. In our algorithm the probabilistic property of each image is used in the relevance feedback process. This property contains the conditional probability of each attribute value given the image and can be updated on the fly by users' feedbacks. It describes a single decision boundary through the features space. Another key idea is to treat positive and negative examples in the feedback differently in the query refinement process, as positive examples often are semantically similar, while negative examples are not. Experiment shows that the performance is better than [6] [8]. Based on this algorithm a system "*iFind*" has been implemented. It shows higher accuracy and effectiveness with experimental results on real-world image collections.

The rest of the paper is organized as follows. In Section 2, we first introduce the Bayesian classifier; then, we describe in detail the proposed algorithm. Section 3 shows our user interface and the experimental results over real-world images. Conclusions are drawn in Section 4.

2. The Proposed Learning Methods

2.1. Bayesian Classifier

Consider vector x in R^n that obeys Gaussian distribution; then, the probability density function of x is[3]:

$$p(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\varepsilon)^T \Sigma^{-1} (x-\varepsilon)} \quad (1)$$

where $x = [x_1, ..., x_n]$, $\varepsilon = [\varepsilon(x_1), ... \varepsilon(x_n)]$, and $\Sigma = \varepsilon\{(x-u)(x-u)^T\}$.

We can get the following Bayesian decision boundary function that is the probability of x belong to the *i*th class w_i [3]:

$$g_i(x) = \lg P_i(x) = -\frac{1}{2}(x-\varepsilon_i)^T \Sigma_i^{-1} (x-\varepsilon_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln |\Sigma_i| + \ln P(w_i) \quad (2)$$

2.2. Positive Feedbacks

We apply the Bayesian classifier to deal with the feedback process in CBIR. Each image belongs to an unknown semantic class. As shown in **Fig. 2**, our system is a example-based image retrieval system. That is, a user provides system an example image as a query and the system retrieves similar images from the image database. It is impossible that the low level feature of the example image is just at the distribution center of a semantic class of images. Our idea is to construct a Bayesian classifier for query image by its positive examples. The parameter of this classifier can be considered as the real 'query' of this image and could be updated by more feedbacks. Hence, the proposed method is both a query refinement and a weight updating process.

It is reasonable to assume that the distribution of image features follows Gaussian distribution. Fig. 1 shows the statistics of the color histogram features from a real-world image set, which could be considered Gaussian.

Each image P_k can be represented by a vector $\vec{x}_k = [\overrightarrow{x_{k1}}, ..., \overrightarrow{x_{km}}]$ in the feature space where $\overrightarrow{x_{ki}} = [x_{ki1}, ..., x_{kin_i}]$. For each feature vector $\overrightarrow{x_{ki}}$, we use a $n_i \times n_i$ dimension covariance matrix Σ_{ki} and an *n* dimension mean vector ε_{ki} to describe their query vector. n_k is the number of positive feedbacks to image P_k. Since we do not care about the inter-feature covariance, we use a diagonal matrix *diag{* $\sigma_{ki}{}^2$*}* instead, where σ_{ki} (m) = Σ_{ki} (m,m). This is because the inter-feature correlation cannot be estimated accurately and reliably, especially when there are not enough feedbacks.

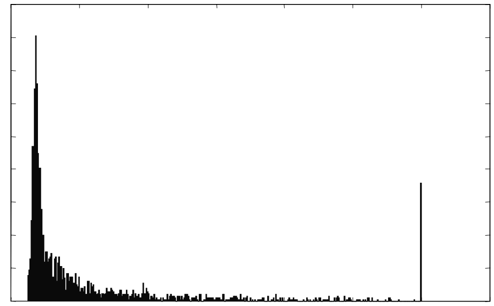

Fig. 1: The distribution of the 7th dimension of feature vector of color histogram in HSV space.

Here is our method to deal with positive feedbacks:

1. Feature Normalization: It puts equal emphasis on each component. For $\overrightarrow{x_{ki}}$ the normalized vector is $\overrightarrow{x_{ki}}' = [x'_{ki1}, ..., x'_{kin_i}]$, where $x'_{ki_m} = \frac{x''_{ki_m} - \varepsilon(x''_{ki_m})}{3\sigma(x''_{ki_m})}$ and $x''_{ki_m} = \frac{x_{ki_m} - \min(X_{ki_m})}{\max(X_{ki_m}) - \min(X_{ki_m})}$. If x_{ki_m} satisfies the Gaussian distribution, it is easy to prove that the probability of x'_{ki_m} being in the range of [-1,1] is 99%.
2. Initialization: Initialize σ_{ki} to be null and let ε_{ki} = $\overrightarrow{x_{ki}}$, $n_k = 1$.
3. Feedback and Update Parameters: In each cycle of P_k's retrieval process, suppose there is a positive example set $C_p = \{P_{p_1} \ldots P_{p_q}\}$: according to equation (2) we have the following updating procedure:
$$\sigma_{ki}{}^2 = n_k\sigma_{ki}{}^2 + \frac{n_k q \varepsilon_{k_i}{}^2 - 2n_k\varepsilon_{k_i}\sum P_{p_i}}{n_k+q} + \sum P_{p_i}{}^2 - \frac{(\sum P_{p_i})^2}{n_k+q}, \quad \varepsilon_{ki} = \frac{n_k \times \varepsilon_{ki} + sum(C_p)}{n_k+q},$$
$$n_k = n_k + q.$$
4. Distance Calculation: For each image P_i in the database, we calculate its distance $d_{i,k}$ to the example image P_k using (2) in the retrieval after the feedback. That is, the similarity of each image in the database to the refined query is determined by (2) based on the positive examples. Notice that $\frac{d}{2}\ln 2\pi, -\frac{1}{2}\ln|\sum_i|, \ln P(w_i)$ in (2) are all constants. The equation is similar to that in the work by Rui and Huang's [8]. This implies that their work conforms to Bayesian classifier. However we use it to construct a Bayesian classifier of the sample image by its positive examples. Our method can keep the former feedbacks, thus the learning has memory and is progressive. Experimental results demonstrate that our method is outperforms the method by Rui and Huang [8].
5. Sorting by distances if have no negative feedbacks.

2.3. Negative Feedbacks

Many algorithms, e.g. Rui and Huang's [6], use the same method to handle negative and positive feedbacks. However, in this work, they are treated differently. Positive examples are usually considered to belong to the same semantic class and there are well agreed-upon understandings. On the other hand, negative examples are often not semantically related. From our observation, negative examples are often *isolated* and *independent*, thus, need to be treated different than positive examples.

We use the following method to deal with the negative examples. Suppose that there is a set of negative feedbacks, $C_n = \{ P_{n1} \ldots P_{nl} \}$, for image P_k. For each element in C_n, a 'dibbling' process is applied in calculating the similarity distance of each database images in the refined retrieval. That is, we penalize images near the negative examples by increasing similarity distance $d_{i,k}$ as defined by (3). With this strategy, there will be a peak in similarity distance at each negative example. By extensive simulation, we found that the function can be well approximated by the combination of a series of Gaussian function.

$$d_{i,k} = d_{i,k} + \sum_{i=1}^{l} (p_{P_{n_i}}(P_i) \times d_{k,n_i}) \qquad (3)$$

where $p_{P_{n_i}}(x)$ is defined in (1) with $\varepsilon = P_{n_i}$, $\Sigma = I$. In this way, images in the database that are close to the negative examples are push away from being selected into the processing retrieved image list.

3. Experimental Results

We have experimentally evaluated the proposed feedback approach with a large test set of images using the *iFind* image retrieval system developed at Microsoft Research.

iFind has a simple interface to search images by examples. Users can select an example image from the database or from file directories. Retrieval results are returned as a ranked list shown in the right image viewer, ordered from left to right and top to bottom. In the retrieval process, the user can provide feedbacks by clicking the ' √ ' or '×' button below each retrieved image. The interface and retrieval process are shown in Fig. 2.

To test the ability of relevance feedback, we use a few commonly used image features as the effectiveness of our approach is independent to the features. **Table 1** shows the features we used in our system.

The image set we used is the Corel Image Gallery. We randomly select 1,600 images from the gallery. These images are manually classified into 32 categories. Each category contains 50 images. The images of the same category as the query example are considered as relevant to the query. Positive feedback process is running as following: images from the same category as the query example that are ranked in top 50 are assigned as the positive feedbacks. For negative feedback process the first two irrelevant images are assigned as negative examples.

Fig. 2: Retrieval and feedback interface of *iFind*. Example image is shown on the up-left banner.

We have compared our work with [6][8]. The comparison results show in

Table 2, **Table 3** and **Table 4**. The accuracy is defined in equation (4):

$$R = \frac{\text{relevant ones retrieved in window}}{\text{window size}} \qquad (4)$$

As shown in tables above, if there is no feedback, our algorithm is not as good as the method presented in [8], but better than that presented in [6]. If there is feedback, our algorithm yields the best performance after one iteration and onwards.

Table 1: Low level features used in '***iFind***'

	Low level features
1	Color histogram in HSV space with quantization 256
2	First and second color moments in Lab space
3	Color coherence vector in LUV space with quantization 64
4	MRSAR texture feature
5	Tamura coarseness histogram
6	Tamura directionality

Table 2: retrieval accuracy without relevance feedback

	Top10	Top20	Top30	Top40	Top50	Top100
R&H[6]	0.031	0.031	0.121	0.151	0.160	0.142
R&H[8]	0.432	0.362	0.317	0.284	0.258	0.183
iFind	0.430	0.362	0.316	0.283	0.258	0.182

Table 3: retrieval accuracy after 1st relevance feedback

	Top10	Top20	Top30	Top40	Top50	Top100
R&H[6]	0.231	0.201	0.187	0.172	0.181	0.153
R&H[8]	0.650	0.529	0.463	0.415	0.375	0.246
iFind	0.749	0.611	0.529	0.467	0.416	0.268

Table 4: retrieval accuracy after 10th relevance feedback

	Top10	Top20	Top30	Top40	Top50	Top100
R&H[6]	0.251	0.203	0.192	0.176	0.190	0.152
R&H[8]	0.776	0.597	0.512	0.455	0.408	0.264
iFind	0.831	0.737	0.652	0.564	0.483	0.292

Table 5 shows the experimental results after one iteration of negative feedback. Clearly, our strategy on negative feedback increases the retrieval accuracy significantly

Table 5: retrieval accuracy after negative feedback

	iFind	R,H[6]
before feedback	0.749	0.231
after feedback	0.827	0.232
Increase	0.078	0.001

We have also done the experiment on a data set of more than 12,000 images and covers a wide range of more than 100 categories. Our algorithm gives the best performance among the three methods being compared.

4. Conclusion

In this paper we present a new learning algorithm in relevant feedback of CBIR. For positive examples we use Bayesian classifier to describe a single decision boundary through the features space for each image. For negative ones we use a ‘dibbling’ process to deal with them. Compared with other works our algorithm is progressive and efficient. It usually requires much less iterations and thus accelerates the feedback process of image retrieval significantly. Moreover, the accuracy is higher than other algorithms.

5. Acknowledge

The work presented in this paper was performed at Microsoft Research China. Thanks Chunhui Hu, Xingquan Zhu, Jacky Shen, Stan Li and Liu Wen-Yin, who have given us valuable comments and suggestions to this paper.

6. References

[1] Buckley, C., and Salton, G. “Optimization of Relevance Feedback Weights,” in Proc of SIGIR’95.

[2] Cox, I. J., Minka, T. P., Papathomas, T. V. and Yianilos, P. N. “The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments” IEEE Transactions on Image Processing , Jan. 2000.

[3] Duda, R. O., and Hart, P. E. (1973). “Pattern classification and scene analysis.” New York: John Wiley & Sons.

[4] Ishikawa, Y., Subramanya R., and Faloutsos, C., “Mindreader: Query Databases Through Multiple Examples,” In Proc. of the 24th VLDB Conference, (New York), 1998.

[5] Lee, C., Ma, W. Y., and Zhang, H. J. “Information Embedding Based on user’s relevance Feedback for Image Retrieval,” Technical Report HP Labs, 1998.

[6] Rui, Y., and Huang, T. S. “Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval”, IEEE Circuits and systems for Video technology, vol. 8, no.5, 1999

[7] Rui, Y., Huang, T. S., and Mehrotra, S. “Content-Based Image Retrieval with Relevance Feedback in MARS,” in Proc. ICIP97.

[8] Rui, Y., and Huang, T. S. “A Novel Relevance Feedback Technique in Image Retrieval,” ACM Multimedia, 1999.

[9] Salton, G., and McGill, M. J. “Introduction to Modern Information Retrieval,” McGraw-Hill Book Company, 1983.

[10] Shaw, W. M. “Term-Relevance Computation and Perfect Retrieval Performance,” Information processing and Management.

[11] Stone, H. S. and Li, C. S. “Image matching by means of intensity and texture matching in the Fourier domain,” in Proc. IEEE Int. Conf. Image Processing. Santa Barbara. CA. Oct 1997.

[12] Vasconcelos, N., and Lippman, A. “Learning from user feedback in image retrieval systems” NIPS’99, Denver, Colorado, 1999.