Probabilistic Neural Networks Supporting Multi-class Relevance Feedback in Region-based Image Retrieval

ByoungChul Ko, Hyeran Byun
Dept. of Computer Science, Yonsei University, Seoul, Korea
{soccer1, hrbyun}@cs.yonsei.ac.kr

ABSTRACT

The relevance feedback approach is one of widely used image retrieval technique. The main role of relevance feedback is to make the links between human's high level concept and computer's low level feature. Even though, there are several relevance feedback algorithms, some algorithms use ad-hoc heuristics or assume that feature vectors are independent regardless of their correlation. In this paper, we propose a new relevance feedback algorithm using Probabilistic Neural Networks (PNN) supporting multi-class learning. In our approach, there is no need to assume that feature vectors are independent and it permits system to insert additional classes for detail classification. In addition, it does not take long computation time for training, because it has only four layers. In PNN classification process, we keep user's entire past feedback actions as history in order to improve performance for future iterations. By history, our approach can capture the user's subjective intension more precisely and prevent retrieval performance from fluctuating or degrading in the next iteration. To validate the effectiveness of our feedback approach, we incorporate the proposed algorithm to our region-based image retrieval tool, FRIP (Finding Region In the Pictures). The efficacy of our method is validated using a set of 3000 images from Corel-photo CD.

1. Introduction

In many Content-Based Image/Video Retrieval (CBIVR) systems, the relevance feedback is one of widely used retrieval model. This model is usually used to interactively improve retrieval performance and allow progressive refinement of query results according to the user's specification.

In case of retrieval systems without relevance feedback [1,2,3], these cannot effectively model high-level concepts and user's subjective perception. For example, if the user wants to search a 'red car', a retrieval system that uses a low-level feature (e.g., color, texture) may only look for rectangle shapes with a red color. In this case, even though the user does not satisfied with the retrieval results, there is no way to obtain a single or more next nearest neighbors. However, in the relevance feedback-based approach [4,5,6], the retrieval is interactively processed between human and computer. The main role of relevance feedback is to establish the link between high-level concepts and low-level features from the user's feedback [6]. In these systems, the user only needs to mark images that he or she thinks similar to the query. By the user's feedback action, weights for similarity are readjusted or query points are moved.

In this paper, we propose a new relevance feedback algorithm using Probabilistic Neural Networks (PNN) supporting multi-class learning. In our system, the user menu for feedback is consisted of four levels: very similar, similar, dissimilar, and very dissimilar. After feedback action, weights for each feature are adjusted according to the user's subjectivity and these are used as weights of PNN. Then, we incorporate the proposed relevance feedback algorithm to our region-based image retrieval tool, **FRIP** (Finding Region In the Pictures) [9].

This paper is organized as follows. In Section 2, we review related work on relevance feedback algorithm. Section 3 provides an overview of our FRIP system. Our proposed relevance feedback algorithm is discussed in Section 4. Then, Section 5 shows experimental results, exploring the performance of our technique using real-world data. Finally, concluding remarks are given in Section 6.

2. Related Work

Most of the existing relevance feedback algorithms are classified into two approaches: query-point movement and re-weighting [8].

Query-point movement method tries to improve the estimate of the "ideal query point" by moving it towards good example points and away from bad example points. However, in this method, the setting of three parameters needs ad-hoc heuristics.

The second approach is based on re-weighting [7]. This approach is also applied to MARS system. Here, if the variance of the good examples is high along the *j-th* axis, apparently any value on the j-axis is acceptable to the user, and therefore the *j-th* axis should have a low weight w_j. A low variance indicates that these relevant images are consistent in this feature and that the feature should be assigned a relatively high weight. Conversely, a high variance gives a relatively small weight by $w_j = 1/\sigma_j$. Even though working reasonably well, the MARS technique is also based on ad-hoc heuristics and did not have a solid theoretical foundation. To improve the limitations of MARS, the MindReader [8] is proposed. Even though, this approach tries to avoid ad-hoc heuristics and develop a mathematical framework for learning of weights, it fails to analyze the working conditions and faces many difficulties in reality [10]. In addition, providing scores to retrieval images places burden on the user.

Others attempts have been studied for years using probabilistic method for estimating of weights [5], Bayesian inference [11,12], decision tree [13], and support vector machines [14].

In this paper, we propose a new relevance feedback algorithm using Probabilistic Neural Networks supporting four levels user selection: very similar, similar, dissimilar, and very dissimilar. In our algorithm, system does not need to assume or know the prior information on training data. Furthermore, our algorithm can provide flexible query results regardless of complexity of query image by using multi-class and history of previous feedback actions.

3. Overview of FRIP System

This system includes our robust image segmentation scheme using a circular filter and description of five features. From the segmented image, we need to extract features from each region. Features (color, texture, normalized area (NArea), location and shape) of each region are used to describe the content of an image and normalized as soon as features are extracted from segmented regions using Gaussian normalization method.

Feature distances between the query and segmented database regions are then estimated. A full description of FRIP system is described in [4, 9].

4. Probabilistic Neural Networks for Multi-class Relevance Feedback

In this paper, we apply Probabilistic Neural Networks (PNN) to our relevance feedback system with additional improvements. In case of PNN, there is no need to assume that feature vectors are independent and it allows the system to insert additional classes for detail classification. Furthermore, it does not take long computation time for training, because it has only four layers.

4.1 Probabilistic Neural Networks

The network structure of PNN is similar to back-propagation. The main difference is that the sigmoid activation function is replaced by one of class of functions, which include, in particular, the exponential. The key advantages of PNN is as follows:

- Training requires only a single pass
- The shape of the decision surfaces can be made as complex as necessary, or as simple as desired, by choosing the appropriate value of the smoothing parameter.
- Sparse samples are adequate for network performance.

The key disadvantage of PNN is that all training sample must be stored and used in classifying new patterns. However, because memory is inexpensive and dense, storage should not be a problem [15, 16].

Figure 1 shows the architecture of PNN. A PNN consists of *d* input units, *n* pattern units, *s* summation units, and *c* output units. Input units comprising input layer is connected to each of the n pattern units. Each pattern unit forms a dot product of the input pattern vector $\boldsymbol{X}$ with a weight vector W_k of pattern unit *k*, $net_k = X \cdot W_k$, and then performs a nonlinear operation on net_k before outputting its activation level to the summation units. The summation units simply sum the pattern units that correspond to the category from which the training pattern was selected. These units are connected to one and only one of *c* output units, as shown in Figure 1.

The PNN is trained in the following way. First, each pattern x of training set is normalized to have unit length ($\sum_{i=1}^{d} x_i^2 = 1$). The first normalized training pattern is placed on the input units. The modifiable weights linking the input units and the first pattern unit are set such that $w_1 = x_1$. Then a single connection from the first pattern unit is made to the output unit corresponding to the known class of that pattern. The process is repeated with each of the remaining training patterns, setting the weights to the successive pattern units such that $w_k = x_k$ for *k*=1, 2,..,n (*k* is the number of pattern units). After such training we have a network that is fully connected between input and pattern units, and sparsely connected from pattern units to output units.

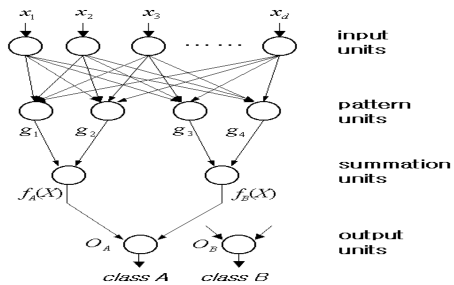

Figure 1. Architecture of PNN having four pattern units and two output units

In this algorithm, we provide the user with four levels relevance feedback such as very similar, similar, dissimilar, very dissimilar. Therefore, each level becomes a pattern unit. Moreover, we input the feature vectors as the feature distances calculated from Section 3. The process of training is as follows:

Initialize $j = 0,\ for\ \ j = 1,...,\ 4;\, i = 1 \ldots 5$

do { $j = j + 1$

$$\bar{x}_{ji} = E[\,j \mid x_{ji}\,] = \sum_{n=1}^{k} x_{ji} \cdot 1(n == j) \,/\, \sum_{n=1}^{k} 1(n == j) \quad (1)$$

$$x_{ji} = \bar{x}_{ji} \,/\, \sqrt{(\textstyle\sum_{i=1}^{5} \bar{x}_{ji}^{2})} \qquad \textit{(Normalize)} \quad (2)$$

$$w_{ji} = x_{ji} \qquad \textit{(Train)}$$

} *while* $(j < 4)$

Here, j represents the number of pattern units (very similar, similar, dissimilar, and very dissimilar) and i represents the number of feature vectors (in this experiment, we use five feature vectors). $\bar{x}_{ji} (= E[\,j \mid x_{ji}\,])$ is an expectation value of x_{ji} which is calculated from k retrieved images and their corresponding pattern marked by the user. Here, *1(·)* is an indicator function. That is, *1(·)* returns 1 if its argument is true, and 0 otherwise.

After the network is trained by setting the weight vector W_k in one of the pattern units, a normalized test pattern $\boldsymbol{X}$ is placed at the input units. Each pattern unit also computes *net activation* (net_k). Then it performs a nonlinear operation on net_k before outputting its activation level to the summation unit instead of the Sigmoid activation function which is commonly used for back-propagation. The nonlinear operation is $\exp(net_k - 1)/\sigma^2$, where σ is a parameter set by the user and determines the width of the effective Gaussian window. Then, connecting the pattern unit's output to the appropriate summation unit. Separate pattern units are required for every training pattern. As indicated in Figure 1, the same pattern units (very similar-similar and dissimilar-very dissimilar) can be grouped by different summation units. The sum of these local estimates give the discriminant function $f_A(X)$ and $f_B(X)$.

The $(-f_A(X) + f_B(X))$ operation gives the desired output unit for the test point.

4.2 Relevance Estimation

The original PNN algorithm is memoryless in that retrieval in previous iterations does not contribute to feature relevance estimates in future iterations. As a result, retrieval performance may fluctuate from one iteration to the next, which might cause performance degradation. To overcome this limitation, we modify discriminat function of PNN by considering the previous history.

We keep user's entire past feedback actions (non-selection-'0', very similar-'1', similar-'2', dissimilar-'3', very dissimilar-'4') in order to improve performance for future iterations. If we are to ignore feedback actions from the previous iterations, rejected images, which were selected as dissimilar in the past, may be retrieved again so that performance at the next iteration may not improve in spite of increase in iteration time. Therefore, we use the previous relevance feedback as *history* to help improve the performance and reduce user's time investment. *History* consists of a pair of both image number and feedback type such as 1, 2, 3, and 4 according to the feedback action.

The modified PNN algorithm supporting multi–class relevance feedback works in the following way.

Initialize $k = 0$, *for* $k = 1,\ldots,4;\ X$: *test pattern*

//input and pattern units part

do { $k = k + 1$

$$net_k = w_k^t X \quad \textit{(Net activation)} \qquad (3)$$

$$g_k = \exp[(net_k - 1)/\sigma^2] \quad \textit{(Activation function)} \qquad (4)$$

$$\textit{if}\ \ history \begin{cases} = 0 & g_k = g_k \\ = 1 & g_k = g_k/\alpha \\ = 2 & g_k = g_k/(\alpha/2) \\ = 3 & g_k = g_k \cdot (\alpha/2) \\ = 4 & g_k = g_k \cdot \alpha \end{cases} \qquad (5)$$

} *while* $(k < 4)$

//summation units part

$$f_A = g_1 + g_2 \quad \textit{(discriminant function)} \qquad (6)$$

$$f_B = g_3 + g_4$$

//output units part

$$class_A = f_A \quad \textit{if}\ (-f_A + f_B) \ge 0 \quad \textit{(similar image)} \qquad (7)$$

$$class_B = f_B \quad \textit{if}\ (-f_A + f_B) < 0 \quad \textit{(non-similar image)}$$

Here, k represents the number of pattern units and g_k represents activation function of each pattern unit. Smoothing parameter σ determines the width of the effective Gaussian window. However, since there is no formal method to determine σ and small changes in σ does not change the misclassfication rate dramatically, we choose σ as 0.6 by experiment. α (α>1) is a decision parameter for *history*. In this paper, we set α to 10. If the query image is saved in past *history*, the output values of activation function are changed by parameter α. f_A is the discriminant function of summation unit for very similar and similar pattern unit. f_B is the discriminant function of summation unit for very similar and dissimilar pattern unit. According to the two discriminant functions, the output, or decision units determine one class between similar image and non-similar class.

From the output class and discriminant function, the final score is estimated, and top k nearest images are displayed in ascending order by the output of corresponding discriminant function.

The main steps of our PNN algorithm supporting four levels relevance feedback are shown below:

1. **Let *i* be the current query. Calculate five feature distances after feature normalization.**
2. **Initialize the weight vector W_j to 1.**
3. **Compute *k* nearest images using W_j and weighted linear combination of feature distances.**
4. **User marks *n* images as very similar, similar, dissimilar, or very dissimilar.**
5. **Initiate *history*.**
6. **While [user is not satisfied] Do**
   - **(a) Classify marked *n* images as four classes according to the user's action and save them into buffer for training.**
   - **(b) Estimates W_j from training process using training data in buffer.**
   - **(c) Update new weights W_j.**
   - **(d) Compute the value of discriminant function using new weights and *history*.**
   - **(e) Determine whether target images are classified class A or class B by Equation (7).**
   - **(f) Compute *k* nearest images of class A using the value of discriminant function by ascending order.**
   - **(g) User marks the *n* images as four levels.**
   - **(h) Update *history*.**

5. Experimental Results

In this experiment, we choose all user constraints (color, texture, scale, shape, location-care) and top 20 nearest neighbors are returned that provide necessary relevance feedback. After *k* images are retrieved, the user marks the images as very similar (oo), similar (o), dissimilar (x) or very dissimilar (xx) according to the user's subjectivity. By these feedback actions, weights for feature distances are adjusted toward to the user's perception. After the user's selection, five weights for feature distance are re-adjusted by training process.

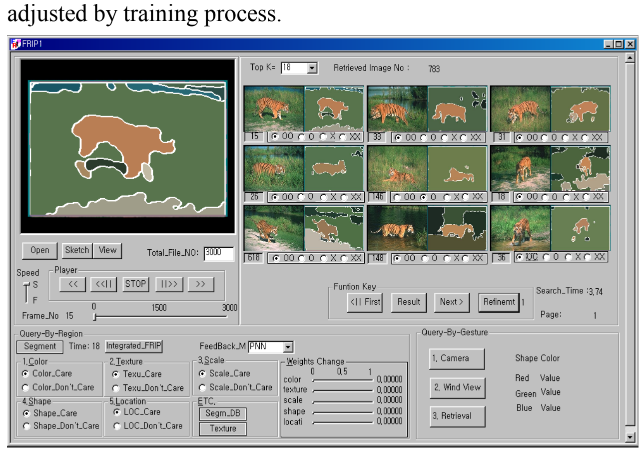

Figure 2. System interface of FRIP supporting four levels relevance feedback

The goal here is to retrieve images from the database that include region similar to the query region. The experiments are carried out on 9 specific domain data such as two kinds of Sun, tiger, car, two kinds of eagles, airplane, traffic sign and flower. We have performed experiment using a set of 3000 images from Corel-photo CD, covering a wide variety of content ranging from natural images to graphic images without pre-selected categories. To validate the effectiveness of our feedback approach, we compare the retrieval performance of our algorithm with PFRL [5], Modified version of PFRL [4], Yong's feedback algorithm [6], and Rocchio's formula [7]. Figure 3 shows performance comparison for each feedback algorithm evaluated by precision using PNN, PFRL, MPFRL, Yong's method, and Rocchio's formula. As we can see, in case of our proposed PNN, it shows better performance without oscillating or decreasing compare to other feedback algorithms because it uses *history* and supporting four levels user selection.

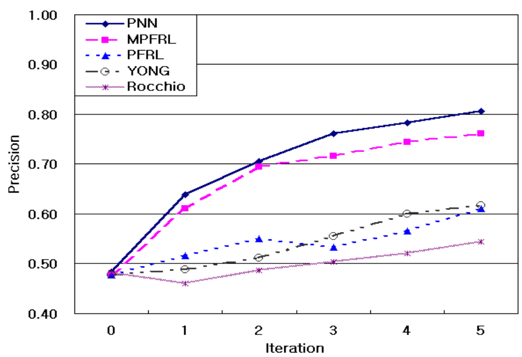

Figure 3. The performance comparison for each feedback algorithm evaluated by precision using PNN, MPFRL, PFRL, Yong's method, and Rocchio's formula.

These results are reported in Table I. As we can see from the Table I, our approach consistently increases the performance at each query. The average precision of FRIP is about 48% without relevance feedback. However, after 5 iterations, the average precision of FRIP is increased to 81%.

Table I. Retrieval performance on top 20 retrieved images according to Iteration Time (IT).

Query	IT-0	IT-1	IT-2	IT-3	IT-4	IT-5
Sun-1	0.45	0.60	0.75	0.90	0.95	1.00
Eagle-1	0.45	0.75	0.75	0.90	0.90	0.95
Airplane	0.20	0.50	0.60	0.60	0.65	0.70
Tiger	0.55	0.65	0.80	0.80	0.80	0.80
Car	0.35	0.50	0.55	0.70	0.70	0.75
Flower	0.90	1.00	1.00	1.00	1.00	1.00
Eagle-2	0.55	0.65	0.80	0.80	0.80	0.80
Sun-2	0.45	0.70	0.75	0.80	0.80	0.80
Traffic sign	0.40	0.45	0.45	0.45	0.50	0.50
Average	**0.48**	**0.64**	**0.71**	**0.77**	**0.79**	**0.81**

6. Conclusion

In this paper, we introduce a new relevance feedback algorithm based on Probabilistic Neural Networks (PNN) supporting four levels relevance feedback learning. At the training process, weights for feature distances are adjusted according to the feature variation and the user's specification. Then, at the classification process, weights are applied to each pattern unit and k nearest neighbors are returned according to the *history* and discriminant function. In this approach, there is no need to assume that feature vectors are independent and it permits system to insert additional classes for detail classification. Furthermore, it does not take long computation time for training, because it has only four layers. Furthermore, since it provides four levels relevance feedback, it can capture the user's intension more precisely and it can improve performance for future iterations by recording all users' feedback actions. The experimental results using real image data show that our approach can indeed rapidly improve the region-based retrieval performance of an image database system.

Acknowledgments

This research was supported as a Brain Neuroinformatics Research Program sponsored by the Korean Ministry of Science and Technology (M1-0107-00-0008).

7. Reference

[1] Faloutsos, M. Flickner, W.Niblack, D. Petkovic, W. Equitz, R. Barber. "Efficient and Effective Querying by Image Content", *Research Report #RJ 9203 (81511), IBM Almanden Research Center, San Jose,* 1993.

[2] Carson, M.Thomas, S. Belongie, J.M. Hellerstein, and J. Malik. "Blobworld: A system for region-based image indexing and retrieval", *In Proc. Int. Conf. Visual Inf. Sys,*.1999.

[3] Y. Ribner, L. J. Guibas, C. Tomasi, "The earth mover's distance, multi-dimensional scaling, and color-based image retrieval," *Proceeding of the ARPA Image Understanding Workshop*, pp. 661-668. May 1997.

[4] ByoungChul Ko, JingPeng, and Hyeran Byun, "Region-Based Image Retrieval Using Probabilistic Feature Relevance Feedback," *Pattern Analysis and Application (PAA)*, Vol. 4, 174-184, 2001

[5] Peng, J., Bhanu, B., and Qing, S., "Probabilistic Feature Relevance Learning for Content-Based Image Retrieval," *Computer Vision and Image Understanding*, Vol. 75, No. 1/2, 150-164. 1999

[6] Yong Rui, Thomas S. Huang, Michael Ortega and Sharad Mehrotra, "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 5, 644-655, Sept. 1998.

[7] Joseph John Rocchio, "Relevance feedback in information retrieval," In Gerard Salton, editor, *The SMART Retrieval System-Experiments in Automatic Document Processing*, pp. 313-323, Prentice Hall, Englewood Cliffs, N.J., 1971.

[8] Ishikawa, Y., Subramanya R., and Faloutsos, C., "Mindreader: Query Databases through Multiple Examples," *In proceeding of the 24th VLDB Conference*, NewYork, 1998.

[9] ByoungChul Ko, Hae-Sung Lee, Hyeran Byun, "Region-based image retrieval system using efficient feature description," *The fifteenth International Conference on Pattern Recognition (ICPR00)*, Vol. 4, 283-286, 2000.

[10] Yong Rui and Tomas Huang, "Optimizing Learning in Image Retrieval," *IEEE CVPR2000*, June, 2000.

[11] Nuno Vasconcelos and Andrew Lippman, "Bayesian Relevance Feedback for Content-Based Image Retrieval," *IEEE Workshop on Content-based Access of Image and Video Libraries*, 63-67, 2000.

[12] Christophe Meilhac and Chahab Naster, "Relevance Feedback and Category Search in Image Databases," *IEEE Int. Conference on Multimedia Computing and Systems*, 512-517, 1999.

[13] Sean D. MacArhur, Carla E. Brodley, and Chi-Ren Shyu, "Relevance Feedback Decision Trees in Content-Based Image Retrieval," *IEEE Workshop on Content-based Access of Image and Video Libraries*, 68-72, 2000.

[14] Pengyu Hong, Qi Tian, Huang, T.S., "Incorporate support vector machines to content-based image retrieval with relevance feedback," *IEEE Int. Conference on Image Processing*, 750-753, 2000.

[15] Donald F. Specht, "Probabilistic Neural Networks and Polynomial Adaline as Complementary Techniques for Classification," *IEEE Transaction on Neural Networks*, Vol. 1, 111-121, March 1990.

[16] Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification*, A Wiley-Interscience Publication, Second Edition, 2000