# A RETRIEVAL PATTERN-BASED INTER-QUERY LEARNING APPROACH FOR CONTENT-BASED IMAGE RETRIEVAL

*Adam D. Gilbert, Ran Chang, and Xiaojun Qi*

adg7@hermes.hood.edu
Computer Science Department, Hood College, Frederick, MD 21701-8575
ran.c@aggiemail.usu.edu and Xiaojun.Qi@usu.edu
Computer Science Department, Utah State University, Logan, UT 84322-4205

## ABSTRACT

This paper presents a retrieval pattern-based inter-query learning approach for image retrieval with relevance feedback. The proposed system combines SVM-based low-level learning and semantic correlation-based high-level learning to construct a semantic matrix to store retrieval patterns of a certain number of randomly chosen query sessions. User's relevance feedback is utilized for updating high-level semantic features of the query image and each database image. Extensive experiments demonstrate our system outperforms three peer systems in the context of both correct and erroneous feedback. Our retrieval system also achieves high retrieval accuracy after the first iteration.

***Index Terms*—** CBIR, retrieval pattern-based inter-query learning, semantic matrix, semantic features

## 1. INTRODUCTION

Relevance feedback (RF) techniques have been widely used in content-based image retrieval (CBIR) to bridge the semantic gap and improve retrieval performance. However, most RF techniques use intra-query learning (a.k.a., short-term learning) to perform query tuning in a single retrieval session. Recently, inter-query learning (a.k.a., long-term learning) has been employed to further improve retrieval performance. Among these, retrieval pattern-based learning is the most effective. Here, we briefly review several such representative techniques.

Retrieval pattern-based inter-query learning techniques aim to establish the relationship between the current and previous query sessions by analyzing image retrieval patterns. If the two sessions have similar image retrieval patterns, these learning techniques assume that the user must be searching semantically similar images. As a result, images with similar retrieval patterns are always returned as retrieval results. Heisterkamp [1] applies the latent semantic analysis method on the term-by-document matrix to learn a generalized relationship between the current query and the search history. He *et al.* [2] use the semantic space to store retrieval patterns of all the query sessions and find semantically similar images. Han *et al.* [3] uses the memory learning technique to form a knowledge memory model to store the semantic information and learn semantic relations. Hoi *et al.* [4] apply the statistical correlation on the retrieval log to analyze the relationship between the current and past retrieval sessions. However, all these learning techniques require a matrix of size $N \times N$ to store feedback information, where $N$ is the number of images in the database. The matrix may also be sparse if all the queries fall into several semantic categories. This sparsity may deteriorate the learning performance of a large-scale database. Moreover, erroneous feedback may also lead to the storage of incorrect semantic information and degrade retrieval performance.

In this paper, we propose a novel retrieval pattern-based inter-query learning scheme for CBIR. First, we apply SVM-based low-level classification and semantic correlation-based high-level classification in intra-query learning to perform query tuning in a single retrieval session. Second, we build a semantic matrix in inter-query learning to store retrieval patterns (i.e., labels of relevant and irrelevant images) of a certain number of randomly chosen query sessions. Third, we extract semantic features of the database images based on the historical feedback gathered from the semantic matrix. Fourth, we update semantic features of the query image by reinforcing semantically relevant features and suppressing semantically irrelevant features using semantic features of the relevant and irrelevant images selected by the user during RF iterations. Fifth, we apply the semantic correlation-based technique to measure the similarity between updated semantic features of the query and semantic features of each database image to return top retrieved images. The rest of the paper is: Section 2 presents our proposed learning approach. Section 3 compares our system with three peers. Section 4 draws conclusions and presents future directions.

## 2. PROPOSED LEARNING SCHEME

Our proposed CBIR system consists of four components: 1) Initial retrieval: Perform low-level visual feature-based image retrieval when the semantic matrix is empty (i.e., no semantic relationship between images is stored). 2) Intra-query learning: Apply SVM-based low-level classification and semantic correlation-based high-level classification in each iterative RF process of a query session to perform query tuning. 3) Inter-query learning: Use the vector space model [5] to construct a semantic matrix to store retrieval patterns of past query sessions. 4) Online retrieval: Apply the semantic correlation-based technique on updated semantic features of the query and semantic features of each database image to find the semantically similar images. The following subsections explain each component in detail.

### 2.1. Initial Retrieval

The initial retrieval uses the low-level feature-based similarity to find top $n$ images for the user to label. We use 64-bin HSV color histogram and 36 features to represent each database image. The 36 features consist of 9 color, 18 edge, and 9 texture components. Color features are the first three moments in HSV color space. Edge features correspond to an 18-bin edge direction histogram of the converted grayscale image. Texture features are the entropy of each of nine wavelet detail subbands of the grayscale image. The Euclidean distance is applied on the normalized features to measure the similarity between the query and each database image.

### 2.2. Intra-Query Learning: A Single Query Session

During the following iterations in a single query session, we employ radial basis function (RBF) kernel-based SVM classification and semantic correlation-based classification to return top retrieved images. SVM classification is to find a better classification boundary to discriminate positive images from negative images in the database. The distance from each database image to the classification boundary can be computed to measure the visual similarity between the query image and each database image. The images with the largest positive distance to the classification boundary are considered as the most similar to the query image. The semantic correlation-based classification is to find the statistical relationship between updated semantic features of the query image and each database image, wherein semantic features are dynamically updated based on the user's RF and the growing semantic matrix. The learning process for any query image $q(t)$ (i.e., the query image $q$ at the $t^{th}$ iteration) consists of the following steps:

1. Perform initial retrieval (section 2.1) to compute the Euclidean distance $EDist_i$ between each database image $I_i$ and the query image $q$.
2. Return top $n$ images based on $EDist_i$.
3. Let the user select relevant (positive) images from the returned pool while treating non-selected images as irrelevant (negative) images.
4. Train SVM using accumulative positive and negative images gathered during RF iterations.
5. Compute the normalized SVM-based distance $SDist_i$ between $I_i$ and the classification boundary.
6. Add $EDist_i$ and $SDist_i$ to measure the low-level feature-based similarity between $I_i$ and $q(t)$.
7. If iteration $t$ equals 1, call InitQuery to initialize semantic features of the query.
   Otherwise, call UpdateQuery to modify semantic features of the query.
8. If query's semantic features contain little information (i.e., the feature elements are mostly 0's), use the low-level feature-based similarity computed in step 6 to return top $n$ images. Otherwise,
   1) Perform the correlation-based operation to compute the similarity between semantic features of $I_i$ and $q(t)$ by:
   $$S_i^{Sem} = I_i^S \cdot q^S(t) = \sum_k I_{i,k}^S q_k^S(t) \qquad (1)$$
   where $q_k^S(t)$ is the $k^{th}$ element of semantic features of query $q$ at the $t^{th}$ iteration, and $I_{i,k}^S$ is the $k^{th}$ element of semantic features of $I_i$.
   2) Combine the low-level feature-based similarity computed in step 6 with the semantic similarity computed by (1) to return top $n$ images.
9. Repeat steps 3 through 9 for three more iterations.

The InitQuery function aims to initialize the query's semantic features using semantic features of the relevant and irrelevant images selected by the users at the first iteration. It starts with finding the semantic rows corresponding to the relevant and irrelevant images. The query's semantic features are initialized as:

$$q_k^S(t) = (s_k^{P,1} \mid \ldots \mid s_k^{P,Np}) \,\&\, \overline{(s_k^{N,1} \mid \ldots \mid s_k^{N,Nn})} \qquad (2)$$

where $s_k^{P,i}$ and $s_k^{N,i}$ are the $k^{th}$ element of semantic features of the $i^{th}$ positive and negative images, respectively. $N_p$ and $N_n$ denote the number of positive and negative images, respectively. We further design customized logical operations, as summarized in Table 1, to effectively involve three values (e.g., 1, 0, and -1) in the computation. The first two columns represent the values of semantic features of the positive images or the values of semantic features of the negative images. The third column represents the results of our logical "or" operation for all nine combinations of three values. The fourth column represents the combined logical "and" results by integrating the resultant value from semantic features of the positive images (represented by X) and the resultant value from semantic features of the negative images (represented

Table 1: Truth table of our customized logical operators

| X | Y | X\|Y | X& $\overline{\text{Y}}$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 0 | -1 | -1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 |
| 1 | -1 | 0 | 1 |
| -1 | 0 | -1 | -1 |
| -1 | 1 | 0 | -1 |
| -1 | -1 | -1 | -1 |

by $\overline{Y}$).

The UpdateQuery function aims to update the query's semantic features by reinforcing semantically relevant features and suppressing semantically irrelevant features using the semantics of the relevant and irrelevant images selected by the user during the remaining iterations (i.e., 2nd, 3rd, and 4th iterations). It updates the query using both semantic matrix, which will be described in section 2.3, and the user's RF by the following criteria:

$$q_i^S(t+1)=\begin{cases}\alpha q_i^S(t) & (s_i^P=1 \ or \ s_i^N=-1), q_i^S(t)\neq 0\\ 1 & (s_i^P=1 \ or \ s_i^N=-1), q_i^S(t)=0\\ q_i^S(t) & (s_i^P=0 \ or \ s_i^N=0)\\ q_i^S(t)/\beta & (s_i^P=-1 \ or \ s_i^N=1)\end{cases} \quad (3)$$

where $q_i^S(t+1)$ is the $i^{\text{th}}$ element of updated semantic features of query $q$, $s_i^P$ and $s_i^N$ are gathered from the semantic matrix and correspond to the $i^{\text{th}}$ element of semantic features of the positive and negative images, respectively. The parameters α and β are the adjustment rate for positive and negative learning and set to be 1.1 and 2, respectively. We set α to be larger than β so semantically irrelevant features will be suppressed more.

### 2.3 Inter-Query Learning: Build the Semantic Matrix

Our system uses the vector space model [5] to represent the semantic relationships learned from user's RF. In this model, we define an $N\times C$ semantic matrix to store semantics (i.e., retrieval patterns) learned during the RF sessions, where $N$ equals to the number of images in the database and $C$ equals to the number of queries used for learning. Fig. 1 demonstrates an example of the semantic matrix after three queries on a database of eight images. Specifically, each row stores the semantic information about each database image and each column stores the learned semantic information (i.e., hidden high-level features) for a particular query. For example, the first row indicates that the 1st image has the same semantics as the 1st query and does not have any semantic information of the 3rd query. We can infer the meaning of each column in a similar manner. Once the semantic matrix has been populated in the training process, each row contains more accurate semantic information. As a result, we use each row as semantic features to represent the semantic information of the corresponding image.

| | | |
|---|---|---|
| 1 | 0 | -1 |
| 1 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 0 | 1 |
| -1 | 1 | 0 |
| 0 | 1 | -1 |
| -1 | -1 | 0 |
| 0 | -1 | 0 |

Fig.1: An example of the semantic matrix after three queries on a database of eight images. Suppose the user gives two positive and two negative feedback for each query. Positive and negative feedback is labeled as 1 and -1, respectively. For those images which are neither positive nor negative, we label 0 at the corresponding row.

The algorithmic view of constructing the semantic matrix is shown in Fig. 2. During this learning period, we randomly select 10% of the database images from each semantic category as training query images to find as many semantically relevant and irrelevant images as possible using users' RF. We also return 50 images at each of the four iterations of the query session to provide more feedback information. In order to speed up the initial learning and maximize the amount of the semantic relationship information that could be learned on the training set, we ensure that any retrieved image would not be returned in the following iterations.

1. Set the semantic matrix to be empty.
2. Randomly select 10% of the database images from each semantic category as training query images.
3. For each training query image $q_i$
   3.1 Apply intra-query learning (Section 2.2) to retrieve top $n$ images for each of the four iterations.
   3.2 Record all the positive and negative examples labeled at each feedback process.
   3.3 Add a new column to the semantic matrix such that the elements corresponding to the rows of all positive and negative examples are respectively set to 1 and -1, and the remaining elements are set to 0.

Fig. 2: The algorithm of constructing the semantic matrix.

### 2.4 Online Retrieval

After training, we perform online retrieval using the remaining 90% of the database images as queries. No additional learning is performed, and images are allowed to be returned multiple times during a single query session. At each iteration, we return 25 images since they can be easily fit into one screen for users to provide their feedback information. For initial retrieval, we directly use query's semantic features (i.e., the row in the semantic matrix that corresponds to the query) to find semantically similar database images using (1). For the following iterations, we update semantic features of the query using UpdateQuery function and find semantically similar database images using (1).

## 3. EXPERIMENTAL RESULTS

We tested our CBIR system on two data sets: the 2000-Flickr database and the combined 2000-Flickr and 6000-COREL database. The COREL database contains 60 distinct categories with 100 images per category. The Flickr database contains 20 distinct categories with 100 images per category, which are manually picked from the Flickr's API search results.

We first designed three experiments on the 2000-Flickr database. We also designed an automatic feedback scheme to construct the semantic matrix. A retrieved image is considered as positive if it belongs to the same category as query. The retrieval accuracy is computed as the ratio of the positive images to the total returned images. We randomly chose 2%, 5%, and 10% of database images as queries to construct three semantic matrices, respectively. Another three experiments were performed to incorporate the possible erroneous feedback in the real-world RF process, wherein erroneous feedback may result from user inherent subjectivity or laziness. Fig. 3 shows the average retrieval accuracy for 1800 images using different semantic matrices as a learning base. It clearly shows the retrieval accuracy is improved when more training images are used to construct the semantic matrix and therefore more semantic relationships among the images are learned. The retrieval accuracy on the largest learning base is above 90% after the 1st iteration and the 2nd iteration in the context of correct and erroneous RF, respectively. Therefore, we chose 10% of the database images as the training images to construct the semantic matrix for future online retrieval.

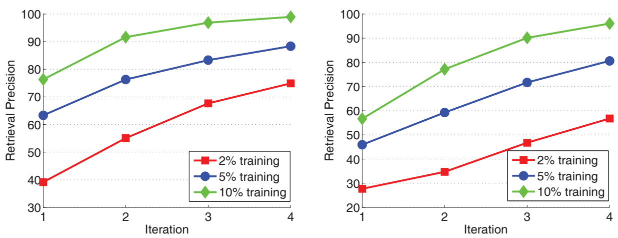


Fig. 3: Retrieval performance on the 2000-Flickr database using different training images and using correct (left) and 5% erroneous RF (right).

We compared our system with Hoi's log-based [4], Han's memory learning [3], and manifold systems [6] on the two data sets. Fig. 4 and Fig. 5 show the average retrieval precision of four systems on the 2000-Flickr database and the combined database after using 10% of the database images to build their perspective learning bases, respectively. It clearly shows that our system achieves the best precision in the context of correct and erroneous RF. Specifically, when comparing to the 2nd best system in the context of correct RF for the last two iterations on the 2000-Flickr and the combined database, our system makes 10.0% and 4.9%, and 14.3% and 7.3% improvement, and achieves accuracy of 96.87% and 98.95%, and 89.31% and 91.87%, respectively. When comparing to the 2nd best system in the context of erroneous RF for the last two iterations on the 2000-Flickr and the combined database, our system makes 27.7% and 18.9%, and 17.9% and 10.6% improvement, and achieves accuracy of 90.14% and 96.04%, and 70.18% and 76.49%, respectively. Our online retrieval time is comparable with three peer systems mainly due to the simplicity of dot operation and semantic feature update.

## 4. CONCLUSIONS

We propose a novel retrieval pattern-based inter-query learning approach for CBIR with RF. Our major contributions consist of:

- Apply both SVM-based low-level classification and semantic correlation-based high-level classification in intra-query learning to perform query tuning.
- Apply the vector space model to build a semantic matrix in inter-query learning to store retrieval patterns.
- Extract semantic features of database images using the semantic matrix and update semantic features of the query using user's RF.
- Apply the correlation-based technique on semantic features to retrieve top semantically similar images.

Experimental results show the proposed system outperforms peer systems and achieves highest retrieval accuracy in all iterations in the context of correct and erroneous feedback. Other forms of the semantic matrix will be investigated in future research.

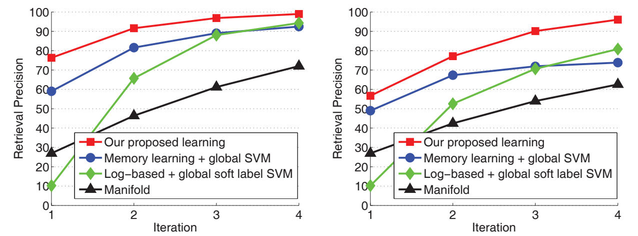


Fig. 4: Comparison of four systems on the 2000-Flickr database using correct (left) and 5% erroneous RF (right).

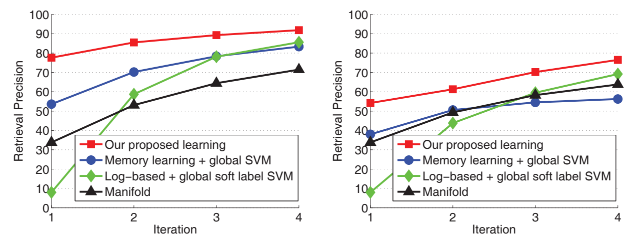


Fig. 5: Comparison of four systems on the combined database using correct (left) and 5% erroneous RF (right).

## 5. REFERENCES

[1] D. R. Heisterkamp, "Building a Latent Semantic Index of an Image Database from Patterns of Relevance Feedback," *Proc. of Int. Conf. on Pattern Recognition*, pp. 134-137, 2002.

[2] X. He, O. King, W. Y. Ma, M. Li, and H. Zhang, "Learning a Semantic Space from User's Relevance Feedback for Image Retrieval," *IEEE Trans. CSVT*, Vol. 13, pp. 39-48, 2003.

[3] J. Han, K. N. Ngan, M. Li, and H. J. Zhang, "A Memory Learning Framework for Effective Image Retrieval," *IEEE Trans. Image Processing*, Vol. 14, No. 4, pp. 511-524, 2005.

[4] S. C. H. Hoi, M. R. Lyu, and R. Jin, A Unified Log-Based Relevance Feedback Scheme for Image Retrieval," *IEEE Trans. Knowledge & Data Engineering*, Vol. 18, No. 4, pp. 509-524, 2006.

[5] G. Salton and M. McGill, *Introduction to Modern Information Retrieval*. New York: McGraw-Hill, 1983.

[6] J. He, M. Li, H. Zhang, H. Tong, and C. Zhang, "Generalized Manifold-Ranking-Based Image Retrieval," *IEEE Trans. Image Processing*, Vol. 15, No. 10, pp. 3170-3177, 2006.