

A Latent Semantic Indexing Based Method for Solving Multiple Instance Learning Problem in Region-Based Image Retrieval

Xin Chen¹, Chengcui Zhang¹, Shu-Ching Chen², Min Chen²

¹*Department of Computer and Information Sciences, University of Alabama at Birmingham
{zhang, chenxin}@cis.uab.edu*

²*School of Computing and Information Sciences, Florida International University
{mchen005, chens}@cs.fiu.edu*

Abstract

Relevance Feedback (RF) is a widely used technique in incorporating user's knowledge with the learning process for Content-Based Image Retrieval (CBIR). As a supervised learning technique, it has been shown to significantly increase the retrieval accuracy. However, as a CBIR system continues to receive user queries and user feedbacks, the information of user preferences across query sessions are often lost at the end of search, thus requiring the feedback process to be restarted for each new query. A few works targeting long-term learning have been done in general CBIR domain to alleviate this problem. However, none of them address the needs and long-term similarity learning techniques for region-based image retrieval. This paper proposes a Latent Semantic Indexing (LSI) based method to utilize users' relevance feedback information. The proposed region-based image retrieval system is constructed on a Multiple Instance Learning (MIL) framework with One-class Support Vector Machine (SVM) as its core. Experiments show that the proposed method can better utilize users' feedbacks of previous sessions, thus improving the performance of the learning algorithm (One-class SVM).

1. Introduction

Most of the existing Relevance Feedback (RF) based approaches [2] [4] consider each image as a whole, which is represented by a vector of N dimensional image features. However, the user's query interest is often just one part of the query image i.e. a region in the image that has an obvious semantic meaning. Therefore, rather than viewing each image as a whole, it is more reasonable to view it as a set of semantic regions. In this context, the goal of image retrieval is to find the semantic region(s) of the user's

interest. Since each image is composed of several regions and each region can be taken as an instance, region-based CBIR is then transformed into a Multiple Instance Learning (MIL) problem [5]. Maron et al. applied MIL into natural scene image classification [5]. Each image is viewed as a bag of semantic regions (instances). In the scenario of MIL, the labels of individual instances in the training data are not available, instead the bags are labeled. When applied to RF-based CBIR, this corresponds to the scenario that the user gives feedback on the whole image (bag) although he/she may be only interested in a specific region (instance) of that image. The goal of MIL is to obtain a hypothesis from the training examples that generates labels for unseen bags (images) based on the user's interest on a specific region.

We addressed the above mentioned problem using One-class Support Vector Machine [1] and built up a learning and retrieval framework in our previous work [11]. The framework applies MIL to learn the region of interest from the users' relevance feedback on the whole image and tells the system to shift its focus of attention to that region. In particular, the learning algorithm concentrates on those positive bags (images) and uses the learned region-of-interest to evaluate all the other images in the image database. The choice of One-class Support Vector Machine is based on the observation that positive images are positive in the same way while negative images are negative in their own way. In other words, instead of building models for both positive class and negative class, it makes more sense to assume that all positive regions are in one class while the negative regions are outliers of the positive class. Therefore, we concentrate on positive image regions and try to model them within our framework.

Chen et al. [6] and Gondra [10] use One-Class SVM in image retrieval but it is applied to the image as a whole. In our approach, One-Class SVM is used to model the non-linear distribution of image regions and separate positive regions from negative ones. Each region of the test images is given a score by the evaluation function built from the model. The higher the score, the more similar it is to the region of interest. The images with the highest scores are returned to the user as query results. Our comparative study shows the effectiveness of this framework with a high retrieval accuracy being achieved on average within 4 iterations [11].

In our experiments, we found out that it is highly likely that repetitive or similar queries are conducted by different users. However, the learning mechanism ignores the previously acquired knowledge from users' relevance feedback and treats each query as an independent and brand-new one. This raised a question i.e. how can we make fully use of the relevance feedback information collected across query sessions. In this paper, we explore a Latent Semantic Indexing based method to analyze and extract useful knowledge from feedbacks stored in database access logs. Related work on using database log information can be found in [9] [12] [13]. However, again their works are based on the whole image instead of image regions. Long-term learning techniques which try to propagate the feedback information across query sessions for region-based image retrieval still appear as an open issue. The information discovery strategy uses Singular Value Decomposition in analyzing log information. In this way, data dimensions are reduced with only important information retained and noise data removed. This differentiates our method from the methods proposed in [9] [12] in which log information is used in a more straightforward way. Our experiments demonstrate that the proposed method performs better in extracting useful information from database access logs.

Section 2 introduces One-class Support Vector Machine. In Section 3, the detailed learning and retrieval approach based on One-class Support Vector Machine is discussed. In Section 4, we propose to use Latent Semantic Indexing to extract information from database access log. The overall system is illustrated and the experimental results are presented in Section 5. Section 6 concludes the paper.

2. One-class support vector machine

One-Class classification is a kind of unsupervised learning mechanism. It tries to assess whether a test point is likely to belong to the distribution underlying the training data. In our case, the training set is

composed of positive image samples only. One-Class SVM has so far been studied in the context of SVMs [1]. The objective is to create a binary-valued function that is positive in those regions of input space where the data predominantly lies and negative elsewhere.

The idea is to model the dense region as a “ball” – hyper-sphere. In Multiple Instance Learning (MIL) problem, positive instances are inside the “ball” and negative instances are outside. If the origin of the “ball” is $\vec{\alpha}$ and the radius is r , a point \vec{x}_i , in this case an instance (image region) represented by an 32-feature vector, is inside the “ball” iff $\|\vec{x}_i - \vec{\alpha}\| \leq r$. This is illustrated in Figure 1 with samples inside the circle being the positive instances.

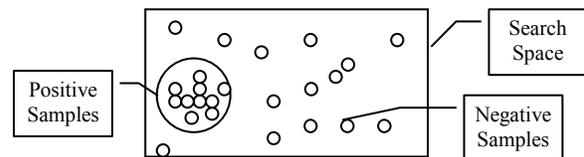


Figure 1. One-class classification

This “ball” is actually a hyper-sphere. The goal is to keep this hyper-sphere as “pure” as possible and include most of the positive objects. Since this involves a non-linear distribution in the original space, the strategy of Schölkopf’s One-Class SVM is first to do a mapping θ to transform the data into a feature space F corresponding to the kernel K :

$$\theta(x_i) \cdot \theta(x_j) \equiv K(x_i, x_j) \quad (1)$$

where x_i and x_j are two data points. In this study, we choose to use Radial Basis Function (RBF) Machine below.

$$K(u, v) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (2)$$

Mathematically, One-Class SVM solves the following quadratic problem:

$$\min_{w, \xi, \rho} \frac{1}{2} \|w\|^2 - \rho + \frac{1}{\alpha n} \sum_{i=1}^n \xi_i \quad (3)$$

subject to

$$(w \cdot \theta(x_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0 \text{ and } i = 1, \dots, n \quad (4)$$

where ξ_i is the slack variable, and $\alpha \in (0,1)$ is a parameter that controls the trade off between maximizing the distance from the origin and enclosing most of the data samples in the “ball” formed by the hyper-sphere. n is the total number of data points in the training set. If w and ρ are a solution to this problem, then the decision function is $f(x) = \text{sign}(w \cdot \theta(x) - \rho)$.

3. Training by one-class SVM

Given a query image, in initial query, the user needs to identify a semantic region of his/her interest. Since no training sample is available at this point, we simply compute the Euclidean distances between the query semantic region and all the other semantic regions in the image database. The smaller the distance, the more likely this semantic region is similar to the query region. The similarity score for each image is then set to the inverse of the minimum distance between its regions and the query region. The training sample set is then constructed according to the user’s feedback.

The user’s feedback provides labels (positive/negative) for a small set of images (e.g., top 30 most similar images). Although the labels of individual instances remain unknown, which is our ultimate goal, we do know the relationship between images and their enclosed regions. If the bag is labeled positive, at least one instance in the bag is positive. If the bag is labeled negative, all the instances in the bag are negative. This is the particular Multiple Instance Learning problem we are trying to solve.

If an image is labeled positive, its semantic region that is the least distant from the query region is labeled positive. In this way, most of the positive regions can be identified. In our experiment, we choose to use Blob-world [3] as our image segmentation method. For some images, Blob-world may “over-segment” such that one semantic region is segmented into two or more “blobs”. In addition, some images may actually contain more than one positive region. Therefore, we cannot assume that only one region in each image is positive. Suppose the number of positive images is h and the number of all semantic regions in the training set is H . Then the ratio of “outliers” in the training set is estimated as:

$$\alpha = 1 - \left(\frac{h}{H} + z \right) \quad (5)$$

z is a small number used to adjust α in order to alleviate the above mentioned problem. Our experiment results show that $z = 0.01$ is a reasonable value.

The training set as well as the parameter α are fed into One-Class SVM to obtain W and ρ , which are used to calculate the value of the decision function for the test data, i.e. all the image regions in the database. Each image region will be assigned a “score” by $w \cdot \theta(x) - \rho$ in the decision function. The higher the score, the more likely this region belongs to the positive class. The similarity score of each image is then set to the highest score of all its regions. It is worth mentioning that except for the initial query in which the user needs to specify the query region in the query image, the subsequent iterations will only ask for the user’s feedback on the whole image.

4. Latent semantic indexing

Latent Semantic Indexing (LSI) was originally used as a mathematical/statistical technique for text mining. It is a novel information retrieval method developed by Deerwester et al. [8]. It is often the case that two documents may be semantically close even if they do not share a particular keyword. The “power” of LSI is that it can find and rank relevant documents even they do not contain the query keywords. The whole procedure is fully automatic.

The fact that LSI does not require an exact match to return useful results fits perfectly with the scenario of image retrieval. Suppose there is a query image – a “tiger” in the grass and the user is interested in finding all images in the image database that contain “tiger”. It is obviously not a good idea to use exact match since no “tiger” image would have exactly the same low-level features with the query image except the query image itself. If we consider an image as a “document”, the “tiger” object is then one of the words in the document. The only difference is that the “tiger” object is not a word, but a multi-dimensional feature vector.

4.1 Constructing “term-document” matrix

The first step in Latent Semantic Indexing is to construct the term-document matrix. It is a 2-D grid with documents listed along the horizontal axis, and content words along the vertical axis. For image retrieval purpose, we will construct a matrix A in a similar sense except that “documents” are images and “content words” are image regions. The matrix has all the training data (user feedbacks) collected by using the method presented in Section 3. Table 1 shows a part of the Matrix.

Table 1 Term-document matrix A

		I_1	I_2	I_3	I_4	I_5	I_6	I_7	..
O_1		0	0	2	2	-2	-2	0	..
O_2		1	1	-1	-1	2	0	0	..
O_3		0	0	0	0	0	1	1	..
...	...								

I_1, I_2, \dots represent images and O_1, O_2, \dots are trained data i.e. image objects. Given an image object O_i , if it is queried by the user, the system depicted in Section 3 will return a series of results upon which user will provide feedback. These feedbacks are collected and stored in the “term-document” matrix. If a returned image I_j is “positive”, the corresponding cell value $A_{ij}(O_i, I_j)$ will be incremented by 1. This corresponds to the situation in LSI-based text mining in which the corresponding cell value in the term-document matrix will be increased by 1 if an exact match of a query keyword is found in a document. If it is “negative”, the value of $A_{ij}(O_i, I_j)$ will be decreased by 1. If its relevance to the query object is unknown (i.e. unlabeled images), the corresponding value in A_{ij} is “0”.

In our experiment, there are 9800 images in the database. Therefore the final matrix A has 9800 columns. The training data are obtained from the log information i.e. retrieval results by the system mentioned in Section 3. User queries and feedbacks are collected continuously over a period of time. The final A matrix has 1245 rows and we normalize it by z-score.

4.2 Singular value decomposition

The key step in Latent Semantic Indexing is to decompose the “term-document” matrix using a technique called Singular Value Decomposition (SVD). LSI works by projecting a large multidimensional space down into a smaller number of dimensions. In doing so, images that are semantically similar will squeeze together. SVD preserves as much information as possible about the relative distance between images while collapsing them down into a much smaller set of dimensions. In doing so, noise data are removed and the latent semantic similarities are revealed.

By singular value decomposition, the “term-document” matrix is first decomposed into smaller

components. Suppose, $A_{m \times n}$ is a “term-document” matrix that has m rows and n columns, the resulting components of A are $U_{m \times n}, S_{n \times n}, V_{n \times n}$ as shown in the figure below.

The columns of U are the left singular vectors which are made up by the eigenvectors of AA^T . V^T has rows that are the right singular vectors, which are made up by the eigenvectors of $A^T A$. S is a diagonal matrix that has the same number of columns as A does. It has the singular values in descending order along the main diagonal of S . These values are square roots of eigenvalues of AA^T or $A^T A$. The SVD represents an expansion of the original data in a coordinate system where the covariance matrix is diagonal.

$$A_{m \times n} = U_{m \times n} \times S_{n \times n} \times V_{n \times n}^T$$

Figure 2. Singular value decomposition of “term-document” matrix

“The beauty of an SVD is that it allows a simple strategy for optimal approximate fit using smaller matrices [8]”. Since the singular values in S are sorted, most important information actually resides in the first k largest values. Therefore, by keeping these k values, we are trying to eliminate noise to the greatest extent. A new matrix, $ASVD$, is then constructed with rank k (see Figure 3).

$$ASVD_{m \times n} = U_{m \times k} \times S_{k \times k} \times V_{n \times k}^T$$

Figure 3. Reduced “term-document” matrix after singular value decomposition

In our experiment, we have a database of 9800 Corel images. They roughly fall into 100 categories. Therefore, the k in our experiment is 100. It is worth mentioning that the estimation of appropriate k for different image databases can be done with the aid of image clustering which is out of the scope of this paper. The detailed experimental results will be shown in Section 5.

4.3 Region based image retrieval by LSI

The matrix $ASVD$, as discussed above, contains the SVD-transformed access log information. The next step is to how to make use of this matrix in our image retrieval process.

In the initial query, we simply compute the Euclidean distance between the query objects and all the objects in the database. The top images are those whose objects have the shortest distances with the query objects and are returned to the user for feedback. User feedbacks are either “positive” or “negative”. In our original system [11], “positive” images are directly fed into One-class SVM for learning purpose. In some cases, the number of “positive” images retrieved by the initial query as small (e.g., 2~3 positive images out of the top 30 images). This lack of positive samples hinders the learning performance of One-class SVM. By using the access log information, we can expect to provide One-class SVM more positive samples.

Given the “positive” images, their relationships with other images in the database with respect to the given query object can be revealed by looking up their corresponding entries in the $ASVD$ matrix. The similarity between two images in the matrix is the dot product of the two column vectors that represent the two images. Since this is object-based image retrieval, we can not simply treat each image as a column vector in full length. Instead only those rows whose objects have a short distance with the query object shall be considered. Therefore, a threshold needs to be set for the distance between the query object and all the other objects along rows of $ASVD$. The purpose is to limit the influence of the trained objects that are far different from the query object in terms of low-level features. In our experiment, we set this threshold to 3. If no object in $ASVD$ can be found to have a distance to the query object less than the threshold, we simply perform the query by using the original system. This capability is extremely important for a real content-based image retrieval system as both short-term learning and long-term learning are desired.

Suppose the query image is I_1 , and the query object is O_q . In Table 2(a), assume that objects O_1 and O_3 are the objects whose distances to O_q are less than the threshold. Therefore, the similarity between I_1 and all the other images in the database with respect to O_q is measured by the dot product of column vectors as shown in Table 2(b). Those images that have similarity values greater than 1 with all the “positive” images are added to the positive image set for One-class SVM. For each of these image samples, its object that has the shortest Euclidean distance to the query object O_q is identified and fed into One-class SVM. The Euclidean

distance is used since the SVM model is not yet available (trained) at this point.

Notice that, in the initial query, we did not simply fetch all the relevant images that have been previously marked positive by users from the log file (Matrix A). Instead, we use the same initial query method as adopted in our original system [11]. We did this simply for comparison purpose. As its initial query mechanism is the same as that of the original system, we can easily measure the effectiveness of the proposed mechanism by examining the performance gains in the following learning and retrieval cycle when log information is used. Then another question may be raised – which matrix is better, in terms of providing useful information to One-class SVM, A or $ASVD$? As shown in our experiment in Section 5, $ASVD$ can provide more useful information regarding positive samples and therefore One-class SVM performs better in the next round. Another advantage of using $ASVD$ is that it does not require exact match with the query object while A does require it. In other words, only those images that have been previously marked positive can be retrieved when using matrix A while using $ASVD$ can discover potentially positive images.

Table 2(a) An example of matrix $ASVD$

	I_1	I_2	I_3	I_4	I_5	I_6
O_1	-1.3	0.5	0.8	7.2	-0.1	-8.5
O_2	1.5	2.5	-0.5	0.05	0.1	-7.5
O_3	-4.3	1.5	0.8	4.3	0.55	20.5
O_4	2.9	3.5	-6.7	0.09	-6.8	-0.3

Table 2(b) Reduced image vectors

	I_1	I_2	I_3	I_4	I_5	I_6
O_1	-1.3	0.5	0.8	7.2	-0.1	-8.5
O_3	-4.3	1.5	0.8	4.3	0.55	20.5

5. Experimentation

5.1 Image segmentation and feature extraction

Figure 4 shows the proposed system. In the first step, images are segmented into semantic regions, with each represented by a 32-feature vector – three texture features, two shape features and 27 color features.

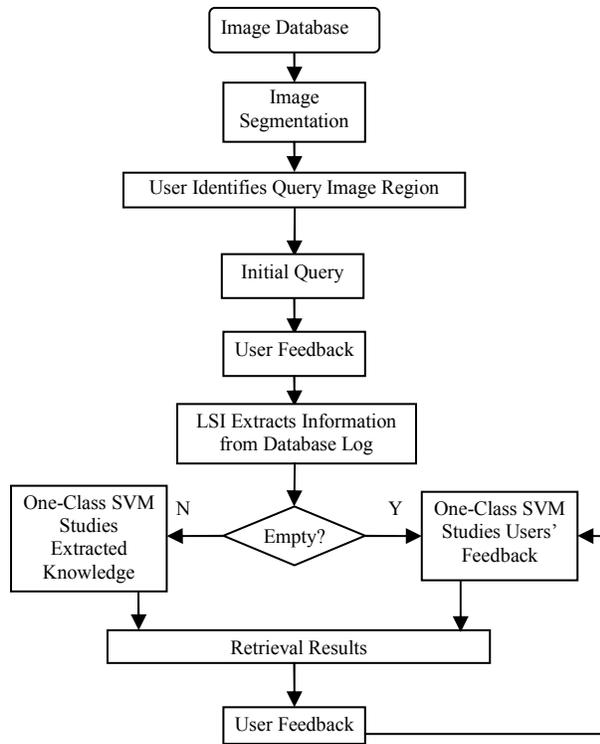


Figure 4. Flowchart of our image retrieval system

After the initial query, user gives feedbacks, which are returned to the system. In the first query, these feedbacks are used to find information from database logs. If useful knowledge is found, our One-Class SVM based algorithm learns from these knowledge otherwise it learns directly from the user’s current feedbacks and the refined retrieval results are returned to the user.

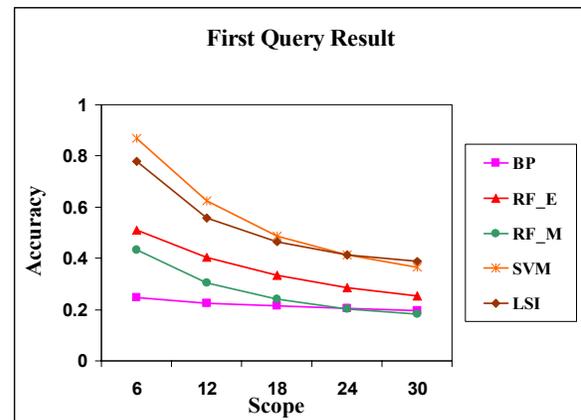
5.2 System performance evaluation

The experiment is conducted on a Corel image database consisting of 9,800 images from 98 categories. After segmentation, there are in total 82,552 image segments. Fifty images are randomly chosen from 20 categories as the query images. In our database log, we have collected altogether 1245 distinct queries.

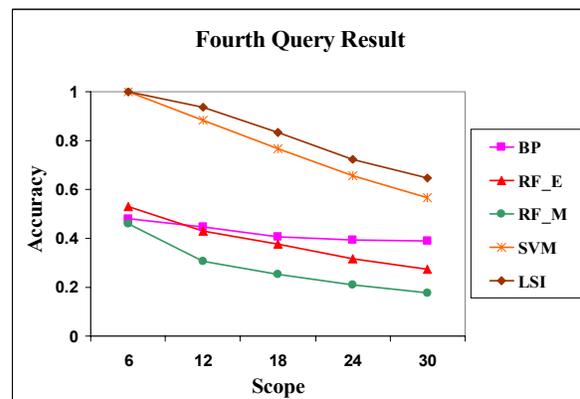
In order to test the performance of LSI, we compare our algorithm with the one that does not consider log information [11]. We also compare the performance of our system with two other relevance feedback algorithms: 1) Neural Network based Multiple Instance Learning (MIL) algorithm with relevance feedback [7]; 2) General feature re-weighting algorithm [2] with

relevance feedback. For the latter, both Euclidean and Manhattan distances are tested.

Five rounds of relevance feedback are performed for each query image - Initial (no feedback), First, Second, Third, and Fourth. The accuracy rates within different scopes, i.e. the percentage of positive images within the top 6, 12, 18, 24 and 30 retrieved images, are calculated. Figure 5(a) shows the result from the First Query while Figure 5(b) shows the result after the Fourth Query. “BP” is the Neural Network based MIL which uses both positive and negative examples. “RF_E” is feature re-weighting method with Euclidean Distance while “RF_M” uses Manhattan Distance. “LSI” is the proposed system and “SVM” refers to the same retrieval mechanism except that database log information is used in the retrieval process [11].



(a)



(b)

Figure 5. (a) Retrieval accuracy after the 1st query; (b) retrieval accuracy after the 4th query

It can be seen from Figure 5 that the accuracy of the proposed algorithm outperforms all the other 3 algorithms. Especially, the proposed algorithm shows

better performance than “SVM” – the one that does not consider log information. It also can be seen that the Neural Network based MIL (BP), although not as good as the feature re-weighting method (RF_E and EF_M) in the First Query, demonstrates a better performance than that of general feature re-weighting algorithm after 4 rounds of learning.

In order to answer the question raised in Section 4 i.e. which matrix is better, in terms of providing useful information to One-class SVM, A or ASVD, we further compare the retrieval result using A and ASVD. Figure 6 shows the fourth round retrieval result of a “horse” region using A. Figure 7 shows the fourth round retrieval result of the same region using ASVD. As shown in Figure 6 and 7, the leftmost image is the query image. This image is segmented into 8 semantic regions (outlined by red lines). User identifies the “horse” region as the region of interest (the 3rd image from left outlined by a blue rectangle). For this specific query object, the retrieved result using ASVD is better than that using A.



Figure 6. A sample retrieval result by using matrix A

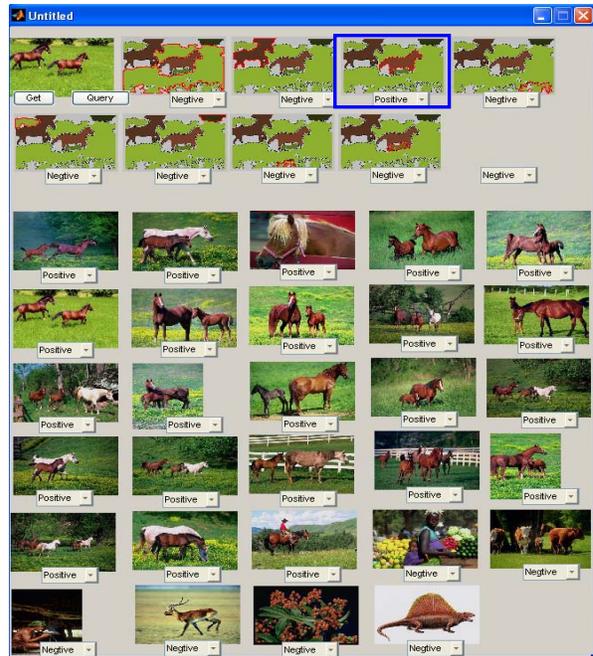


Figure 7. A sample retrieval result by using matrix ASVD

Figure 8 shows the average result after the fourth query of the 50 query images using matrix A and ASVD. On average, the performance of the latter is also consistently better than the former.

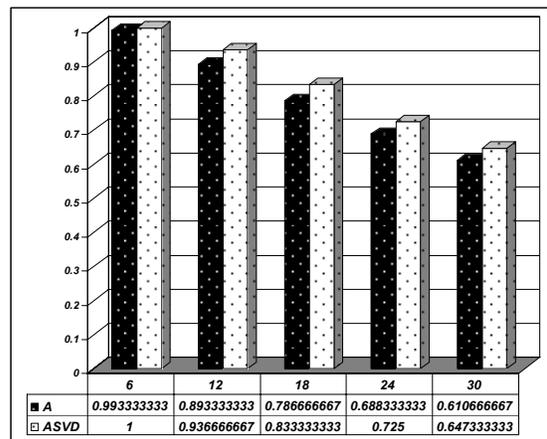


Figure 8. Comparison between A and ASVD on the fourth query

In Figure 9, the accuracy rates of our algorithm across 5 iterations are illustrated. Through each iteration, the number of positive images increases steadily.

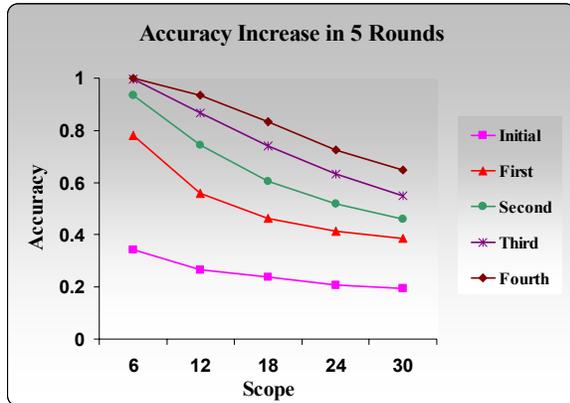


Figure 9. Retrieval Results of “LSI” across 5 Iterations

6. Conclusion

In this paper, we propose a RF-based MIL framework for single region based CBIR system. We propose a method to make use of the previously acquired feedbacks for region-based image retrieval. The method uses Latent Semantic Indexing which can fully exploit the feedback information and its effectiveness is demonstrated by experiments. We also adopt One-Class SVM in the image retrieval phase. A particular advantage of the proposed system is that it targets on both short-term learning and long-term learning for region-based image retrieval, which is desired by contemporary CBIR systems since the user is often interested in only one region in the image. The proposed work also transfers the One-Class SVM learning for region-based CBIR into a MIL problem. Due to the generality of One-Class SVM, the proposed system can better identify user’s real need and remove the noise data.

7. Acknowledgement

The work of Chengcui Zhang was supported in part by SBE-0245090 and the UAB ADVANCE program of the Office for the Advancement of Women in Science and Engineering.

8. References

[1] B. Schölkopf., J.C. Platt et al., “Estimating the Support of a High-dimensional Distribution”, *Microsoft Research Corporation Technical Report MSR-TR-99-87*, 1999.

[2] Y. Rui., T. S. Huang, and S. Mehrotra, “Content-based Image Retrieval with Relevance Feedback in MARS”, *Proc. of the International Conference on Image Processing*, 1997

[3] C. Carson, S. Belongie, H. Greenspan, and J. Malik, “Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying”, *IEEE Trans on Pattern Analysis and Machine Intelligence*, Vol. 24, No.8, 2002.

[4] Z. Su., H. J. Zhang, S. Li, and S. P. Ma, “Relevance Feedback in Content-based Image Retrieval: Bayesian Framework, Feature Subspaces, and Progressing Learning”, *IEEE Trans on Image Processing*, Vol. 12, No. 8, 2003.

[5] O. Maron and T. Lozano-Perez, “A Framework for Multiple Instance Learning”, *Advances in Natural Information Processing System 10*, MIT Press, 1998.

[6] Y. Chen, X. Zhou, S. Tomas., and T. S. Huang, “One-Class SVM for Learning in Image Retrieval”, *Proc of IEEE International Conference on Image Processing*, 2001.

[7] X. Huang, S.-C. Chen, M.-L. Shyu and C. Zhang, “User Concept Pattern Discovery Using Relevance Feedback and Multiple Instance Learning for Content-Based Image Retrieval”, *Proc of the 3rd International Workshop on Multimedia Data Mining (MDM/KDD’2002)*, 2002.

[8] S. Deerwester, S. T. Dumais, T. K. Landauer, G. Furnas and R. Harshman, “Indexing by Latent Semantic Analysis”, *Journal of the American Society of Information Science*, October, 1990.

[9] C.-H. Hoi and M. R. Lyu, “A Novel Log-based Relevance Feedback Technique in Content-based Image Retrieval”, *Proc of the 12th annual ACM international conference on Multimedia.*, New York, U.S.A., October 2004.

[10] I. Gondra and D. R. Heisterkamp, “Adaptive and Efficient Image Retrieval with One-Class Support Vector Machines for Inter-Query Learning”, *WSEAS Transactions on Circuits and Systems*, Vol. 3, No. 2, April 2004.

[11] C. Zhang, X. Chen, M. Chen, S.-C. Chen, and M.-L. Shyu, “A Multiple Instance Learning Approach for Content Based Image Retrieval Using One-Class Support Vector Machine”, *Proc. of IEEE International Conference on Multimedia & Expo (ICME)*, Amsterdam, the Netherlands, 2005.

[12] M.-L. Shyu, S.-C. Chen, M. Chen, and C. Zhang, “Affinity Relation Discovery in Image Database Clustering and Content-based Retrieval”, *ACM Multimedia 2004 Conference*, New York, USA, October 2004.

[13] J. Fournier ad M. Cord, “Long-term similarity learning in content-based image retrieval”, *ICIP 2002*, Rochester, New-York, September 2002.