

# Building a Latent Semantic Index of an Image Database from Patterns of Relevance Feedback

Douglas R. Heisterkamp  
Department of Computer Science  
Oklahoma State University  
Stillwater, OK 74078  
doug@cs.okstate.edu

## Abstract

*This paper proposes a novel view of the information generated by relevance feedback. Latent semantic analysis is adapted to this view to extract useful inter-query information. The view presented in this paper is that the fundamental vocabulary of the system is the images in the database and that relevance feedback is a document whose words are the images. A relevance feedback document contains the intra-query information which expresses the semantic intent of the user over that query. The inter-query information then takes the form of a collection of documents which can be subjected to latent semantic analysis. An algorithm to query the latent semantic index is presented and evaluated against real data sets.*

## 1. Introduction

Many recent approaches to content base image retrieval rely upon relevance feedback to improve query results [7, 6, 8]. The user is presented with a set of images. To refine the retrieval set, the user assigns a relevant/irrelevant measure to each image and requests a new retrieval set. Often, this process is conducted in the vector space model. The query is represented as a point in a feature space. The retrieval set consists of the set of images whose feature vectors are the  $k$  nearest neighbors of the query point. After learning from the relevance feedback, the system generates a new query location and adjusts the distance metric for the next  $k$  nearest neighbors search. This process iterates until the user is satisfied with the image retrieval or abandons the search. The system is using the *intra-query* information from relevance feedback to increase performance.

Relevance feedback also provides *inter-query* information that most current systems do not exploit. The difficulty in exploiting the inter-query information is due to the

unknown intent of the user. Each user, with each query, could have a different model of the semantic information with which he is searching the image database.

Müller, *et al.*, presented an approach of using the inter-query information [4]. The historical logs of user interaction with the *Viper* [9] system were used to adjust the inverse document frequency weighting of the features. The discriminating features over the historical data received an increase to their weight while the non-discriminating features received a reduction to their weight. In the vector space model, this would correspond to adjusting the initial distance metric from the typical Euclidean distance to a distance metric that fits the historical data (discriminant analysis).

Müller's approach uses the inter-query information to adapt the feature space of the images. Images are composed of features and the features are the basic components of information. The approach proposed in this paper takes a fundamentally different view. The images in the database are the basic terms of information. The intra-query information from relevance feedback is a document whose words are the images of the database and whose meaning expresses the semantic intent of the user. The inter-query information takes the form of a collection of documents which can be subjected to latent semantic analysis.

## 2. Latent Semantic Indexing

Latent semantic indexing, LSI, [2] is an important technique in information retrieval. It is an effort to overcome the problems of *synonymy* and *polysemy* when using individual words to retrieve documents. That many words may have the same meaning is the problem of synonymy. That a word may have many meanings is the problem of polysemy. To overcome these problems, LSI uses the context of the word's usage to uncover the hidden (or latent) meaning of the word. The context of a word's usage, in a broad sense,

is the entire document which contains the word.

LSI creates a semantic space by applying singular value decomposition, SVD, to the term-by-document matrix,  $\mathbf{M}$ . Each column of  $\mathbf{M}$  represents a document. The components of the column represent the relationship of the term to the document (such as a frequency weight of the occurrences of the term in the document). The term-by-document matrix is then approximated by using the  $k$  largest singular values and their associated singular vectors:

$$\underbrace{\mathbf{M}}_{t \times d} = \underbrace{\mathbf{U}}_{t \times r} \underbrace{\mathbf{S}}_{r \times r} \underbrace{\mathbf{V}^T}_{r \times d} \approx \underbrace{\hat{\mathbf{M}}}_{t \times d} = \underbrace{\hat{\mathbf{U}}}_{t \times k} \underbrace{\hat{\mathbf{S}}}_{k \times k} \underbrace{\hat{\mathbf{V}}^T}_{k \times d} \quad (1)$$

where  $t$  is number of terms,  $d$  is number of documents,  $r$  is rank of  $\mathbf{M}$ ,  $\mathbf{U}$  and  $\mathbf{V}$  are orthonormal, and  $\mathbf{S}$  is diagonal. The  $\hat{\mathbf{M}}$ ,  $\hat{\mathbf{U}}$ ,  $\hat{\mathbf{V}}$  and  $\hat{\mathbf{S}}$  matrices are the approximations of the respective matrices when using just the  $k$  largest singular values. These matrices can be used to compare terms,  $\hat{\mathbf{M}}\hat{\mathbf{M}}^T = \hat{\mathbf{U}}\hat{\mathbf{S}}^2\hat{\mathbf{U}}^T$ , and to compare documents,  $\hat{\mathbf{M}}^T\hat{\mathbf{M}} = \hat{\mathbf{V}}\hat{\mathbf{S}}^2\hat{\mathbf{V}}^T$ . To process a previously unknown query document, first a *pseudo-document*,  $\mathbf{T}_q$ , is created as a vector of its component terms. This vector is then projected into the semantic space by  $\mathbf{F}_q = \hat{\mathbf{U}}^T\mathbf{T}_q$ . The distance of the query to each of the documents is then the distance of  $\mathbf{F}_q$  to the corresponding column of  $\hat{\mathbf{S}}\hat{\mathbf{V}}^T$ .

### 3. LSI for Image Databases

We view the images of of a database as the fundamental vocabulary of the system. An image may have many semantic meanings (polysemy) and many images have have similar meaning (synonymy). Just as with words, the semantic meaning of images is defined in context of other images. The pattern of relevance feedback from a query is a document composed of many terms (images). The terms of an individual document are assumed to have a latent semantic relationship.

With these assumptions, the creation and use of an LSI for an image database requires some modifications to the typical use of an LSI. Typically in LSI, you are given a query document of known terms and asked to determine a set of similar documents. In our case, we are given an unknown term and asked to determine a set of similar terms. To answer that question, the process outlined in Figure 1 is used.

In step 0, the term-by-document matrix,  $\mathbf{M}$ , is created from historical patterns of relevance feedback. The non-zero term components of a document have a positive or negative value depending on whether the feedback of the corresponding image was relevant or irrelevant. All columns are normalized to unit length. SVD is used to determine  $\hat{\mathbf{U}}$ ,  $\hat{\mathbf{S}}$ , and  $\hat{\mathbf{V}}$ . Note that  $\mathbf{M}$  is sparse as a single column will generally have only 40-100 non-zero values.

**Figure 1. Latent Semantic Indexing of Images**

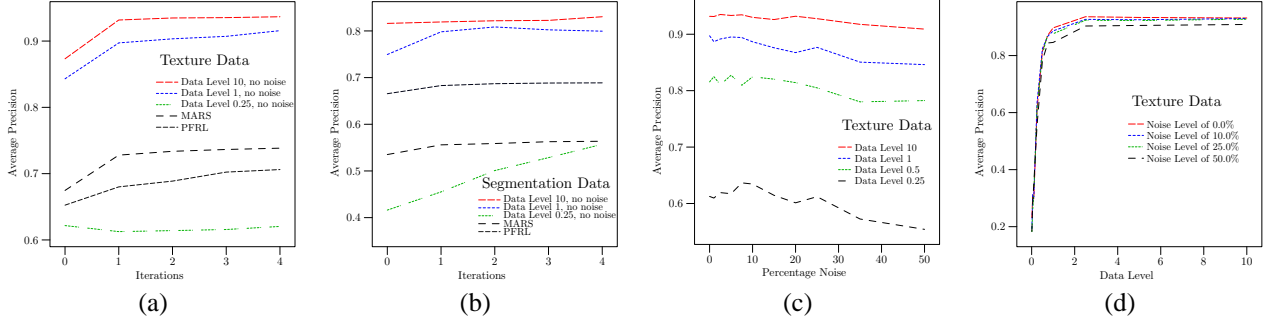
0. Create  $\hat{\mathbf{M}} = \hat{\mathbf{U}}\hat{\mathbf{S}}\hat{\mathbf{V}}^T$ .
1. Input unknown query image and Create initial pseudo-document,  $\mathbf{T}_q$ .
2. Project into latent semantic space,  $\mathbf{F}_q = \hat{\mathbf{U}}^T\mathbf{T}_q$ .
3. Using  $\mathbf{F}_q$ , Select set of terms (images).
  - a. Find K-NN documents to  $\mathbf{F}_q$
  - b. Select most probable terms and most informative terms from the documents of step 4a and place in  $R$ , the retrieval set.
4. Return the retrieval set,  $R$ , to the user.
5. **If** relevance feedback is available **then** create pseudo-document,  $\mathbf{T}_q$ , from terms in  $R$  **else** record session and **exit**
6. GoTo step 2.

In step 1, the initial pseudo-document is created for the query image. Since the image is an unknown term, we create the initial pseudo-document by falling back to a feature space of images. The  $K$  nearest neighbor images of the query is found and a positive relevance weighting is assigned to the corresponding term location in the pseudo-document. The relevance weighting can be any similarity measure. The pseudo-document,  $\mathbf{T}_q$ , is normalized to unit length.

The query's location,  $\mathbf{F}_q$ , in the semantic space is used to select the select the terms to place in the retrieval set,  $\mathcal{R}$ , in step 3. The proposed method is to find a set documents similar to  $\mathbf{F}_q$  in the semantic space and select the most probable and most informative terms of those documents. The approach of using the most probable and the most informative terms is similar in concept to [10]. The set of most probable terms exploit our current knowledge to retrieve the most relevant images. The set of most informative terms provide an exploration to improve our knowledge of the user's query.

The most probable terms are determined from a weighted sum of the terms associated with the  $K$  nearest neighbor documents of  $\mathbf{F}_q$  in the semantic space. The relationship of terms to a document,  $d$ , is given by  $d$ th column of  $\hat{\mathbf{M}}$  (or equivalently,  $\hat{\mathbf{U}}\hat{\mathbf{S}}\hat{\mathbf{V}}_d^T$  where  $\hat{\mathbf{V}}_d^T$  is  $d$ th column of  $\hat{\mathbf{V}}^T$ ). The weighting is based on the distance of the document to  $\mathbf{F}_q$ .

The approach used to find the most informative terms is based on maximizing the decrease in *entropy impurity* [3, p. 398] which is an approach used in the construction of decision trees. All of the  $K$ -NN documents are placed in the root node. A term,  $t$ , is selected and used to distribute the documents to the left and right child nodes. The  $t$ th



**Figure 2. (a)-(b) Average Precision vs Iteration of Relevance Feedback, (c) Average Precision vs Noise Percentage, and (d) Average Precision vs Data Level**

term value of a document is thresholded to -1, 0, or 1. The document is placed in the left child if the value is -1, the right child if 1, and both children if 0. The entropy impurity,  $i(N)$ , of node  $N$  is defined as

$$i(N) = - \sum_j P(\omega_j) \log_2 P(\omega_j) \quad (2)$$

where  $P(\omega_j)$  is the frequency ratio of category  $\omega_j$  in node  $N$ . In our case, the  $\omega_j$ 's are the relevant and irrelevant labels of each of the terms. The decrease in entropy impurity from the knowledge of the user's relevance feedback for a term,  $t$ , is

$$\Delta i(N) = i(N) - P_L i(N_L) - (1 - P_L) i(N_R) \quad (3)$$

where  $N_L$  and  $N_R$  are the left and right child nodes, and  $P_L = \frac{|N_L|}{|N_L| + |N_R|}$  is the weighting of the left node with  $|N|$  as the number of documents in node  $N$ . The term causing the greatest decrease in entropy impurity is selected as an informative term. The most informative terms are selected sequentially.

In step 5, if relevance feedback is provided then a new pseudo-document,  $\mathbf{T}_q$ , is created. All components of  $\mathbf{T}_q$  are zero except those corresponding to images labelled by relevance feedback. The non-zero terms of  $\mathbf{T}_q$  are set to a positive or negative value based on the relevance or irrelevance labelling, of the corresponding images of this and all previous relevance feedback iterations of the query.

## 4. Experimental Results

First we compare the image retrieval using LSI with retrieval techniques that do not use inter-query information. Second, we investigate the response of LSI retrieval with respect to quantity (data level) of inter-query information and with respect to quality (noise level) of inter-query information.

The LSI retrieval method is compared with MARS [8] and PFRL [6]. Performance is measured by *average precision*.

**Average Precision:** The non-interpolated average precision is defined as

$$\frac{1}{|\mathcal{R}el|} \sum_{d \in \mathcal{R}el} \frac{|\{d' | d' \in \mathcal{R}el, \text{Rank}(d') \leq \text{Rank}(d)\}|}{\text{Rank}(d)} \quad (4)$$

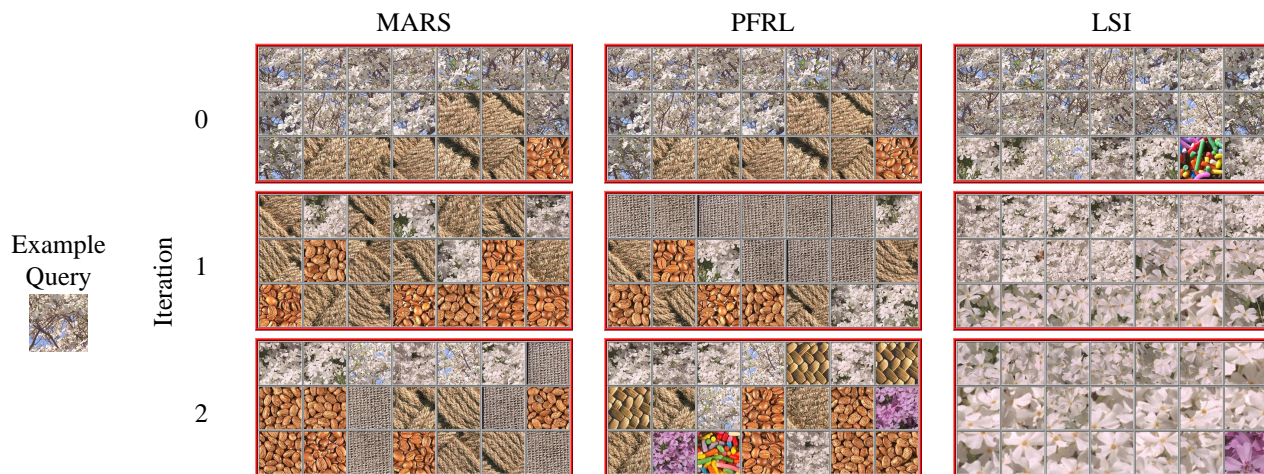
where  $\mathcal{R}el$  is the set of relevant items in the data set, and Rank is an ordering of the data set based on similarity with the query.

The following data sets were used for evaluation: **Texture** — the texture data from MIT Media Lab. A total of 640 images of  $128 \times 128$  with 15 classes are in the database. The images in this database are represented by 8 Gabor filters (2 scales and 4 orientations). **Segmentation** — the segmentation data set, taken from the UCI repository [5], consists of images that were drawn randomly from a database of 7 outdoor images. There are 7 classes, each of which has 330 instances for a total of 2310 images in the database. These images are represented by 19 real valued attributes.

To determine the free parameters, a ten-fold cross-validation was conducted. The values reported are the average of the ten tests. The parameters for MARS and PFRL are optimized. We make no claim on using optimal values for the LSI approach as the parameters were selected after a very coarse sampling.

To create the historical information for LSI, random queries were conducted on the training set using two iterations of relevance feedback on sets of twenty images. The method of retrieval for each queries was randomly selected. In creating  $\mathbf{M}$ , the top 100 singular values were used. The sparse matrix SVD calculation was done with SVDPACKC [1].

The LSI approach was evaluated at different levels of data. The level of data is expressed as the number of documents relative to the number of terms (images). The av-



**Figure 3. Sequential image selection for an example query of MIT Texture Data. The most relevant images returned to the user are selected from images that the user has not seen.**

erage precision over the texture data is presented in Figure 2a. With significant levels of data (1 to 10 times the number of images), the LSI approach performs exceeding well. Without much historical information (1/4 the number of images), performance is less than MARS and PFRL. The same trend holds in Figure 2b which presents the results over the segmentation data.

The response of the LSI approach to poor quality of historical information was investigated by adding simulated noise. When generating the historical relevance feedback documents, the probability that the user’s feedback for each image was flipped is the noise level. The plot of average precision versus noise is presented in Figure 2c. The quality of historical information has a small effect on the quality of the retrieval. The dominating effect on the performance of the LSI approach is the amount of historical information.

Lastly, an example image retrieval is presented in Figure 2d. In this example, the improved MARS [7] is used. As was the complete texture data set. The example presents sequential sets of images to the user (that is, previous images are not reshown to the user).

## 5. Summary

By viewing the images of an image database as the terms of vocabulary and by viewing relevance feedback as an expression of those terms as a document, it is possible to use latent semantic indexing to capture useful inter-query information. Initial investigation suggests that LSI is robust to poor historical information but is dependent on the availability of large amounts of historical information.

## References

- [1] M. W. Berry. SVDPACK: A Fortran-77 software library for the sparse singular value decomposition. Technical Report CS-92-159, University of Tennessee, Knoxville, TN, June 1992.
- [2] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
- [3] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, Inc., New York, second edition, 2001.
- [4] H. Müller, W. Müller, D. M. Squire, S. Marchand-Maillet, and T. Pun. Long-term learning from user behavior in content-based image retrieval. Technical Report 00.04, University of Geneva, 2000.
- [5] P. Murphy and D. Aha. UCI repository of machine learning databases. [www.cs.uci.edu/~mlearn/MLRepository.html](http://www.cs.uci.edu/~mlearn/MLRepository.html).
- [6] J. Peng, B. Bhanu, and S. Qing. Probabilistic feature relevance learning for content-based image retrieval. *Computer Vision and Image Understanding*, 75(1/2):150–164, 1999.
- [7] Y. Rui and T. Huang. Optimizing learning in image retrieval. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina*, pages 236–243, 2000.
- [8] Y. Rui, T. Huang, and S. Mehrotra. Content-based image retrieval with relevance feedback in mars. In *Proceedings of IEEE International Conference on Image Processing, Santa Barbara, California*, pages 815–818, October 1997.
- [9] D. M. Squire, W. Müller, H. Müller, and J. Raki. Content-based query of image databases, inspirations from text retrieval: inverted files, frequency-based weights and relevance feedback. In *11th Scandinavian Conference on Image Analysis*, pages 143–149, Greenland, 1999.
- [10] J. Yoon and N. Jayant. Relevance feedback for semantics based image retrieval. In *International Conference on Image Processing*, pages 42–45, October 2001.