SPARSITY BASED IMAGE RETRIEVAL USING RELEVANCE FEEDBACK

Osman Günay, A. Enis Çetin

Bilkent University
Department of Electrical and Electronics Engineering
06800, Ankara, Turkey

ABSTRACT

In this paper, a Content Based Image Retrieval (CBIR) algorithm employing relevance feedback is developed. After each round of user feedback Biased Discriminant Analysis (BDA) is utilized to find a transformation that best separates the positive samples from negative samples. The algorithm determines a sparse set of eigenvectors by L1 based optimization of the generalized eigenvalue problem arising in BDA for each feedback round. In this way, a transformation matrix is constructed using the sparse set of eigenvectors and a new feature space is formed by projecting the current features using the transformation matrix. Transformations developed using the sparse signal processing method provide better CBIR results and computational efficiency. Experimental results are presented.

***Index Terms*—** Relevance Feedback, CBIR, BDA, L1-ball, Sparsity

1. INTRODUCTION

Relevance feedback is the process of refining the outputs of an information retrieval system based on the input from a user after he/she is presented with initial query results [1]. Relevance feedback is also used in CBIR problems [2].

Relevance feedback and Shannon entropy are used in [3], to obtain a diverse set of refined queries. At each iteration of feedback they choose a number of samples around the query to present to the user for feedback. The points are added to the set using a cost function that is a weighted average of a distance function and an empirical entropy function. They use the Biased Discriminant Analysis (BDA) to evaluate their method and refine the returned query points.

Biased discriminant analysis introduced in [4] can be used to increase the performance of relevance feedback algorithms by efficiently learning from a few training samples. In this method, which is also called (1+x)-*class learning*, the number of classes is not known but only one class is important. The user, during the relevance feedback, marks the samples as “positive” or “negative”. Although there is one positive class, there might be more than one negative class. The main idea of the method is that during relevance feedback, positive samples are more likely to have a compact support, and therefore the method is biased toward positive examples. The solution of the BDA is equivalent to solving a generalized eigenvalue-eigenvector computation.

Another discriminant analysis method is the Common Spatial Patterns (CSP) that is used in brain computer interface (BCI) applications. In [5], they convert the generalized eigenvalue-eigenvector computation to an optimization problem to obtain a sparse solution.

In our relevance feedback application after the user labels the initial query results, a sparse transformation is obtained by projecting the solution of the biased discriminant optimization problem on L1-ball [6]. For each feedback round a new feature space is formed by projecting the current features using the sparse transformation matrix. Sparse solution leads not only to a computationally efficient algorithm but also better CBIR results.

The rest of the paper is organized as follows; in Section 2 we review the BDA algorithm, in Section 3 we introduce the sparse BDA method and in Section 4 we present the experimental results.

2. BIASED DISCRIMINANT ANALYSIS

In BDA two different covariance matrices, A and B, from the positive and negative examples are constructed, respectively. Biased discriminant analysis solves the following Rayleigh Quotient problem:

$$W_{opt} = \arg\max_{W} \frac{|W^T B W|}{|W^T A W|} \tag{1}$$

where

$$A = \sum_{i=1}^{n_a} (a_i - m_a)(a_i - m_a)^T \tag{2}$$

$$B = \sum_{i=1}^{n_b} (b_i - m_a)(b_i - m_a)^T \tag{3}$$

This work was supported in part by the Scientific and Technical Research Council of Turkey, TUBITAK, with grant no. 111E217, in part by European Commission 7th Framework Program with grant number FP7-ENV-2009-1244088 FIRESENSE

a_i and b_i represent the feature vectors extracted from positive and negative samples, respectively, and $m_a = \frac{1}{n_a}\sum_{i=1}^{n_a} a_i$ is the mean of the positive samples. Compared to linear discriminant analysis (LDA) [7], BDA has larger effective dimension (i.e., nonzero eigenvalues of the generalized eigenvalue solution), because the dimension of LDA is equal to "number of classes-1" whereas it is $\min(n_a, n_b)$ for BDA. Therefore BDA usually performs better in separating the positive examples when the number of samples is small [4].

The optimal solution W_{opt} in Eq. 1 is the solution of the generalized eigenvalue problem where the matrix of eigenvectors V and the matrix of eigenvalues Λ satisfy:

$$B_R V = A_R V \Lambda \tag{4}$$

The biased discriminant transform (BDT) matrix can be formed as [4]:

$$T = V\Lambda^{1/2} \tag{5}$$

In a relevance feedback architecture BDA is used as a small sample learner. After each round of feedback BDA transforms the feature space using the BDT matrix in Eq. 5. A major problem with the generalized eigenvalue solution is that it may overfit the data by producing too many small eigenvalues. Instead of solving the generalized eigenvalue problem we employ a sparse signal processing method which yields a significant set of eigenvalues even in small data sets.

3. SPARSE BDA

To obtain a sparse BDT solution the following optimization problem should be solved [5]:

$$w_{opt} = \arg\max_{w} \frac{|w^T B w|}{|w^T A w|} \;\; s.t. \;\; ||w||_1 = z \tag{6}$$

where z determines the sparsity level of the solution. In our method we maximize the Rayleigh Quotient problem subject to the constraint:

$$||w||_1 \leq \alpha \tag{7}$$

which defines an L1-ball. We solve this problem iteratively by making orthogonal projections on L1-ball. Making a projection onto the L1-ball consists of making orthogonal projections onto hyperplanes [8]. The pseudo code of the proposed method is given in Algorithm 1. After solving for W_{opt} we sort the eigenvectors in ascending order in terms of their corresponding eigenvalue amplitudes. We only use the first M eigenvectors. The value of M can be different for each feedback round.

For each eigenvector we find the parameter α that determines the sparsity level of the projection. α can be selected proportional to the L1-norm of each eigenvector v as in the following equation:

$$\alpha = \frac{\sum_i |v(i)|}{(1+\delta)} \tag{8}$$

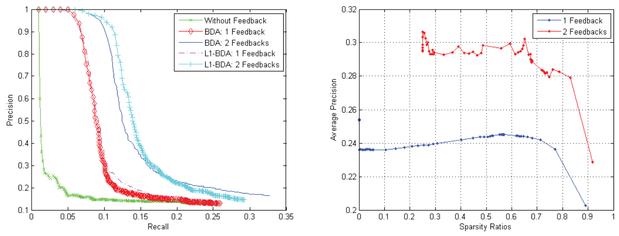
where $\delta \geq 0$ can be used to adjust the sparsity level of the projection, increasing δ yields sparser vectors. The smallest elements of the eigenvector are set to zero depending on the value of α. Since α is different for each eigenvector we cannot determine a single sparsity level for the algorithm, therefore we average the sparsity levels of all eigenvectors to obtain a comparative sparsity level.

Algorithm 1 Sparse BDA Algorithm

```
Solve for $W_{opt}$
Sort eigenvectors in ascending order
Retain only $M$ eigenvectors
for Each eigenvector v do
    Determine $\alpha$
    $N \Leftarrow length(v)$;
    $||v||_1 \Leftarrow \sum_i |v(i)|$;
    for $i = 1 \to N$ do
        $v_k(i) \Leftarrow v(i) + sign(v(i))\frac{(\alpha - ||v||_1)}{N}$;
        if $(sign(v_k(i)) \neq v(i))$ then
            $vk(i) \Leftarrow 0$;
        end if
    end for
    $v_k \Leftarrow \frac{v_k}{\sum_i |v_k(i)|}$;
end for
```

4. EXPERIMENTAL RESULTS

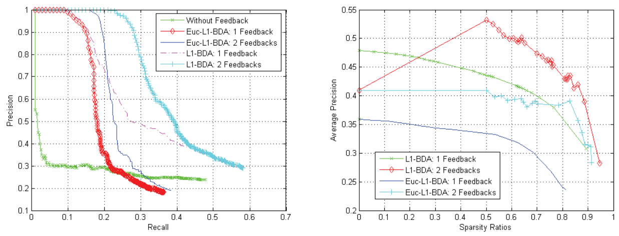
In the experiments we first used a synthetic dataset to test the performance of the sparse BDA algorithm. We created a dataset of 100K vectors; each vector has 64-elements that are distributed according to a Gaussian distribution. The mean of the distribution determines the class of the sample. We created 1000 classes each with 100 vectors. We compare the performances of different methods in terms of precision/recall graphs. We used 10 samples in the dataset as the query vectors and averaged the results to obtain the final precision/recall values. To find the distances between the query and the samples in the dataset we use an exhaustive search method since the dimension of our data is usually high. Indexing methods such as kd-tree [9], work better when the dimension is low. We return 200 vectors for the initial query and assume 50 of them are labeled by the user as positive or negative samples. In Fig. 1a we compare our L1-ball projection based method (called L1-BDA), with the regular BDA after one and two feedback rounds. We observe that the L1-BDA performs significantly better than the regular BDA for both feedback rounds. In Fig. 1b average sparsity ratios that are calculated as the ratio of the number of the zero elements of eigenvectors to the length of the eigenvectors are shown for the Gaussian dataset. Making the vectors too sparse decreases performance, because this means that too few eigenvectors are selected.



(a) (b)

Fig. 1: a) Precision/Recall graph for L1-BDA and BDA on Gaussian dataset. b) Average precision values vs sparsity ratio for the Gaussian dataset.

In Fig. 2a, the proposed projection method is compared with the Euclidean L1-projection method given in [6] on a different realization of the Gaussian dataset (10K classes, 10 samples for each class) used in the first experiment. We see that the proposed method has higher precision values at the same sparsity ratios.

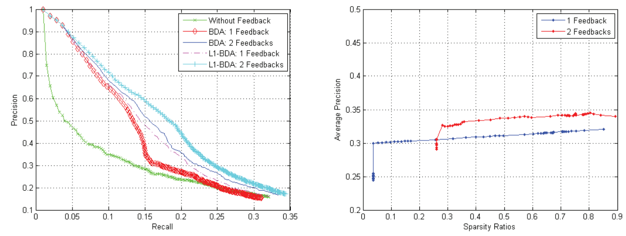


(a) (b)

Fig. 2: The proposed projection algorithm is compared with Euclidean projection method in [6]. In terms of a) precision/recall performance, and b) sparsity ratios.

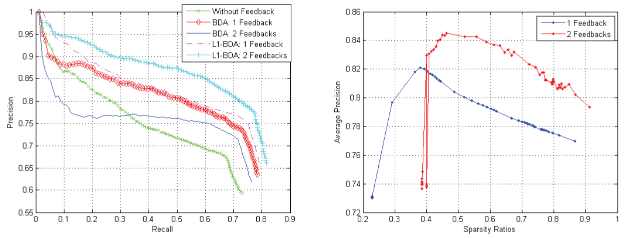
For the second test we used the feature vector set from the AFTER project [10]. There are 3400 samples each with 338 elements corresponding to different color and texture features. The features are obtained from the COREL image dataset; there are 34 classes each with 100 images. We used 34 samples in the dataset as the query vectors and averaged the results to obtain the final precision/recall values. We return 200 vectors for the initial query and assume 50 of them are labeled by the user as positive or negative samples. In Fig. 3a we observe that the L1-BDA performs better than the regular BDA for both feedback rounds. Fig. 3b average sparsity ratios are shown. We see that for this dataset we can have almost % 90 sparsity ratio without decrease in the performance.

For the last experiment we used the KTH-TIPS database that contains 810 images for 10 different classes of colored textures [11]. To extract features from the images we used the



(a) (b)

Fig. 3: a) Precision/Recall graph for L1-BDA and BDA on COREL dataset. b) Average precision values vs sparsity ratio for the COREL dataset.



(a) (b)

Fig. 4: a) Precision/Recall graph for L1-BDA and BDA on KTH-TIPS dataset. b) Average precision values vs sparsity ratio for the KTH-TIPS dataset.

dual-tree complex wavelet transform (DT-CWT) as texture features and histograms in HSV color space as color features. Dual-tree complex wavelet transform tree, is recently developed to overcome the shortcomings of conventional wavelet transform, such as shift variance and poor directional selectivity [12]. To obtain wavelet features we divide images into four non-overlapping blocks and calculate the energies and variances of six different subbands (oriented at +/-15, +/-45, +/- 75) for each block. The combined feature vectors of all blocks are used as the texture feature of the image. We return 100 images for the initial query and assume 50 of them are labeled by the user as positive or negative samples. The results for this test are in Fig. 4a and Fig. 4b.

In Table 1, query response times of BDA and L1-BDA are compared for each feedback round on different datasets. All tests are performed on a PC with Intel I7 3 GHz processor and 6GB ram. D1 has 100K normally distributed samples (the format of the samples is the same as the first experiment) and 10 samples for each class, D2 has 200K total samples and 10 samples for each class, D3 has 100K total samples and 100 samples for each class, D4 has 1M total samples and 100 samples for each class. R1, R2, R3 denote three different feedback rounds. The results are obtained by averaging the response times of different query images. We see that L1-

Table 1: Comparison of query response times of BDA and L1-BDA for each feedback round on different datasets.

		Query Response Times (sec)	
		BDA	L1-BDA
D1	R1	0.2309	0.1508
	R2	0.1274	0.0888
	R3	0.0816	0.0577
D2	R1	1.3202	0.7758
	R2	0.7466	0.4843
	R3	0.4224	0.2964
D3	R1	2.6019	1.5101
	R2	1.4446	0.9156
	R3	0.8174	0.5756
D4	R1	5.2058	2.9993
	R2	2.8516	1.7854
	R3	1.6170	1.1065

BDA usually has lower response times than BDA.

5. CONCLUSION

A method is developed to obtain sparse eigenvectors from the biased discriminant transform by projecting the vectors on L1-ball. The method is used in a relevance feedback framework for CBIR applications. After each round of feedback, features of the images returned by the user are mapped to a new sparse feature space using the sparse transformation. It is possible to achieve high sparsity levels using this method without sacrificing performance. The method performs better than the regular BDA on the colored texture (KTH-TIPS) and object categories datasets (COREL). Since we USE L1-ball projections the method is computationally efficient even on large datasets. Making a projection onto the L1-ball consists of making orthogonal projections onto hyperplanes forming the boundary of the L1 ball.

6. REFERENCES

[1] G. Salton and C. Buckley, “Improving retrieval performance by relevance feedback,” *Journal of the American Society For Information Science*, vol. 41, no. 4, pp. 288–297, JUN 1990.

[2] Y. Rui and T. S. Huang, “A novel relevance feedback technique in image retrieval,” in *Proceedings of the seventh ACM international conference on Multimedia (Part 2)*, New York, NY, USA, 1999, MULTIMEDIA ’99, pp. 67–70, ACM.

[3] C. Dagli, S. Rajaram, and T. Huang, “Leveraging Active Learning for Relevance Feedback Using an Information Theoretic Diversity Measure,” in *Image and Video Retrieval*, vol. 4071, chapter 13, pp. 123–132. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[4] X. S. Zhou and T. S. Huang, “Small sample learning during multimedia retrieval using BiasMap,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001, vol. 1, pp. 11–17.

[5] F. Goksu, N. F. Ince, and A. H. Tewfik, “Sparse common spatial patterns in brain computer interface applications,” in *ICASSP*, 2011, pp. 533–536.

[6] J. Duchi, S. S. Shwartz, Y. Singer, and T. Chandra, “Efficient projections onto the l1-ball for learning in high dimensions,” in *Proceedings of the 25th international conference on Machine learning*, New York, NY, USA, 2008, ICML ’08, pp. 272–279.

[7] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[8] A.E. Cetin and R. Ansari, “Signal recovery from wavelet transform maxima,” *Signal Processing, IEEE Transactions on*, vol. 42, no. 1, pp. 194 –196, jan 1994.

[9] Jon Louis Bentley, “Multidimensional binary search trees used for associative searching,” *Commun. ACM*, vol. 18, pp. 509–517, September 1975.

[10] James C. French, James V. S. Watson, Xiangyu Jin, and W. N. Martin, “Integrating multiple multi-channel cbir systems (extended abstract),” in *Proc. Inter. Workshop on Multimedia Information Systems (MIS*, 2003, pp. 85–95.

[11] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh, “On the significance of Real-World conditions for material classification,” in *European Conference on Computer Vision (ECCV)*, 2004, pp. 253–266.

[12] I.W. Selesnick, R.G. Baraniuk, and N.G. Kingsbury, “The dual-tree complex wavelet transform,” *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 123–151, NOV 2005.