Fusion and Kernel Type Selection in Adaptive Image Retrieval

Anca Doloc-Mihu and Vijay V. Raghavan

The Center of Advanced Computer Studies, University of Louisiana at Lafayette, USA

ABSTRACT

In this work we investigate the relationships between features representing images, fusion schemes for these features and kernel types used in an Web-based Adaptive Image Retrieval System. Using the Kernel Rocchio learning method, several kernels having polynomial and Gaussian forms are applied to general images represented by annotations and by color histograms in RGB and HSV color spaces. We propose different fusion schemes, which incorporate kernel selector component(s). We perform experiments to study the relationships between a concatenated vector and several kernel types. Experimental results show that an appropriate kernel could significantly improve the performance of the retrieval system.

Keywords: fusion, image retrieval, kernel, learning

1. INTRODUCTION

The effectiveness of a CBIR system depends on the choice of visual features and of the similarity metric that models the user perception of similarity. Since the latter is very difficult to model, the current tendency in the Image Retrieval community is to use both content-based image retrieval and text-based image retrieval to enhance the performance of the Image Retrieval System. Learning and result merging in this context have been a concern of both the Information Retrieval community (see[1] for a survey) and the multimedia database community.[2–5]

In a CBIR system, there is a question of how to combine the information derived about an image based on different feature types (e.g., annotation, color, shape, etc.). In this work, we assume that an image is described by multiple feature types, both textual (annotation) and visual (color). In this context, searching for images involves the mixture of these different feature types. The problem is that when using multiple feature types, it is generally not known a priori how these feature types should be integrated in order to determine a combined best decision about the category to which an image belongs.

We are interested in effective and efficient modalities of dynamically learning the weights of the feature types necessary to form a final best ranking of the image collection. Our final goal is building an effective and efficient Web-based Adaptive Image Retrieval System. For this, to reduce the semantic gap, we employ learning via feedback from user. To reduce the curse of dimensionality, we use a kernel based learning method, such as Kernel Rocchio.[6] In this work, we are using vector representations of images in different feature spaces (color, annotation). As retrieval function, we use the inner product function used for text retrieval in Information retrieval.[7]

Since we are dealing with real images, and therefore, with complex queries, it is expected that we need to use non-linear kernels to achieve good retrieval results. Recently, researchers start to seek for ways of automatically selecting the kernel type. Methods like Relevance Vector Machines[8] assume distribution of data, which might fit or not a real collection. Interesting work, which calculates the parameters of the kernel as probabilities, was proposed in.[5] However, for efficiency reasons, our work deals with the kernel selection task in the vector space model.

Our previous work,[9,10] based on color only, revealed interesting insights regarding the relationship between color (in RGB and HSV) and the 12 kernel types investigated. Based on these results, we proposed[10] a method for selecting the kernel type that use score distribution models.

In this work, we are extending our previous work to incorporate more than one feature type. For this, we study different ways the kernel can be used with different fusion schemes. Moreover, we study the applicability (effectiveness and efficiency) of our kernel selection method based on score distribution models when images are represented by multiple feature types.

The goal is to find out the possible relationships between features, fusion schemes and kernel types. To our knowledge, the relationships that might exist between the choice of different kernel type and the fusion process it is not known yet. The final goal is to design the best possible architecture for our system, which we want it to be able to learn dynamically the importance of the different features from user feedbacks at each iteration, and can automatically select the best kernel type to improve this learning process.

For this, in Section 3 we describe our general fusion schemes, which incorporate kernel selector component(s). For simplicity, we present only the case of two feature types, color and annotation. However, the proposed fusion schemes can be further easily adapted to any number of feature type an image retrieval system might be based on.

The paper organization is as follows. Section 2 provides the background. Section 3 describes the proposed fusion schemes. Experiments are presented in Section 4. Finally, Section 5 concludes the paper.

2. BACKGROUND AND PREVIOUS WORK

Let $\mathcal{X}$ = collection of images be our data set, $\mathcal{X} = \{\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_N\}$, $\mathcal{R}$ = the set of the relevant images from the collection and $\mathcal{NR}$ = the set of the non-relevant images. Then, $\mathcal{X} = \mathcal{R} \cup \mathcal{NR}$ and $\mathcal{R} \cap \mathcal{NR} = \mathrm{O}$.

2.1. Vector Concatenation

For simplicity, we assume that our image collection is described by only two feature types, namely color and annotation. Let $\mathrm{P_c}$ be the color vector (here, color histogram in RGB or HSV color spaces) representing the color feature of an image P, and $\mathrm{P_a}$ be the annotation vector representing the annotation description of the same image P. Then, by concatenating[4,11] the two representations of the image P we obtain a new vector $\mathrm{P_{ca}}$ which gives the knowledge about the image:

$$\mathrm{P_{ca}} = (\alpha \mathrm{P_c}, (1-\alpha)\, \mathrm{P_a}), \tag{1}$$

where $\alpha > 0$ represents the weight or the amount of the importance the user give to each of the feature types.

2.2. Retrieval Model

In Image Retrieval, user searches a large collection of images, for instance, that are similar to a specified query. The search is based on the similarities of the image attributes (or features). Here, a linear retrieval form[6,7] similar to the form used for text retrieval matches image queries against the images from collection

$$F : R^N \times R^N \to R, F(\mathrm{P}, \mathrm{Q}) = \mathrm{P}^t\mathrm{Q}. \tag{2}$$

The query image Q contains the features desired by user. The bigger the value of the function F applied to a query Q and an image P, the better the match between the query image Q and the collection image P, or, in other words, the closer the two images.

2.3. Kernel Types

By providing a way for obtaining algorithms working with non-linear decision boundaries from algorithms previously restricted to handling only linearly separable datasets, kernel functions constitutes a very powerful tool, which can enhance the capabilities of a retrieval system based on learning.

In this work, we perform experiments for 6 polynomials and 6 radial basis like kernels,[6,12] with general forms given by the following equations

$$K(\mathrm{x}, \mathrm{y}) = (\langle \mathrm{x}, \mathrm{y} \rangle)^{\mathrm{d}}, \mathrm{d} > 0, \tag{3}$$

and

Table 1. Parameters used for the different kernels.

d	Name
$d=1$	Pol_1
$d=2$	Pol_2
$d=3$	Pol_3
$d=4$	Pol_4
$d=5$	Pol_5
$d=6$	Pol_6

a) polynomials

a	b	c	name
$a=1$	$b=2$	$c=1$	Rad_1
$a=1$	$b=1$	$c=1$	Rad_2
$a=0.5$	$b=2$	$c=1$	Rad_3
$a=0.5$	$b=1$	$c=1$	Rad_4
$a=0.25$	$b=2$	$c=1$	Rad_5
$a=0.25$	$b=1$	$c=1$	Rad_6

b) radials

$$K(\mathrm{x}, \mathrm{y}) = \exp\left(-\frac{\left(\sum_{i=1}^{N} |x_i^a - y_i^a|^b\right)^c}{2\sigma^2}\right), \sigma \in R^+. \tag{4}$$

The values of the parameters (a, b, c, and d) and the names of the kernels used in the experiments are presented in Table 1. In all experiments presented in this work, $\sigma = 1$.

2.4. Kernel Rocchio

Recently, image retrieval systems start using different learning methods for improving the retrieval results. In this work, we use the *Kernel Rocchio* method[6] for learning, which computes the RSVs * of the collection images, as follows:

$$RSV(\phi(\mathrm{P}_k)) = \frac{1}{|\mathcal{R}|}\sum_{\mathrm{P}_i \in \mathcal{R}} \frac{K(\mathrm{P}_i, \mathrm{P}_k)}{\sqrt{K(\mathrm{P}_i, \mathrm{P}_i) \cdot K(\mathrm{P}_k, \mathrm{P}_k)}} - \frac{1}{|\mathcal{NR}|}\sum_{\mathrm{P}_j \in \mathcal{R}} \frac{K(\mathrm{P}_j, \mathrm{P}_k)}{\sqrt{K(\mathrm{P}_j, \mathrm{P}_j) \cdot K(\mathrm{P}_k, \mathrm{P}_k)}}, \tag{5}$$

for any image P_k from the collection. The method combines the simplicity of Rocchio method with the power of non-linear kernel functions to improve the retrieval process.

3. FUSION WITH DIFFERENT KERNEL TYPES

To enhance the performance of the Image Retrieval System we use a learning approach, Kernel Rocchio (Sect. 2.4), based on relevance feedback together with fusion schemes.[1,5,11,13,14] We try to learn dynamically the importance of the different features from users at each iteration. However, to our knowledge, the relationship that might exist between the choice of different kernel types and the fusion process it is not known yet.

In this section we provide different ways the kernel can be incorporated in such fusion schemes. For simplicity, we present only the case of two feature types. However, these schemes can be further easily adapted to any number of feature type an image retrieval system might be based on.

3.1. Fusion with Vector Concatenation

The first fusion scheme that we work with† assumes that the different representations of an image are fused into a concatenated vector, which becomes the new representation of the image (Sect. 2.1). The learning process, which includes the kernel function, is applied on these concatenated vectors.

In this way, the user is no longer required to specify a precise weight α (Eq. 1) for each feature type at the query formulation stage, since they can be learned implicitly by the learning algorithm.[4,11] However, before using a kernel based learning method, the kernel function must be known. For this, we propose to include a "kernel type selector" component in the system. The process is depicted in Figure 1 and requires one level of learning.

*Retrieval Status Value.

†Known also under the name of early fusion.[4]

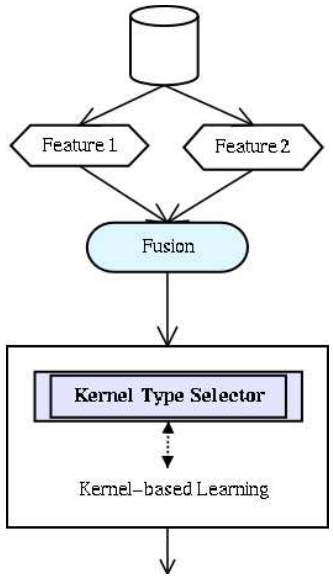


Figure 1. Fusion before learning.

3.2. Fusion without Vector Concatenation

If the concatenated vector is not used, the learning process can be performed under two settings of the parameter α: when α is known (given by user), and when α must be learned (not known).

In the former case, learning can be performed on each feature space separately, and then, the results can be merged[2,3] by using the parameter α. Each feature type have associated its own kernel type selector, which will work only with the knowledge given by the respective feature space.

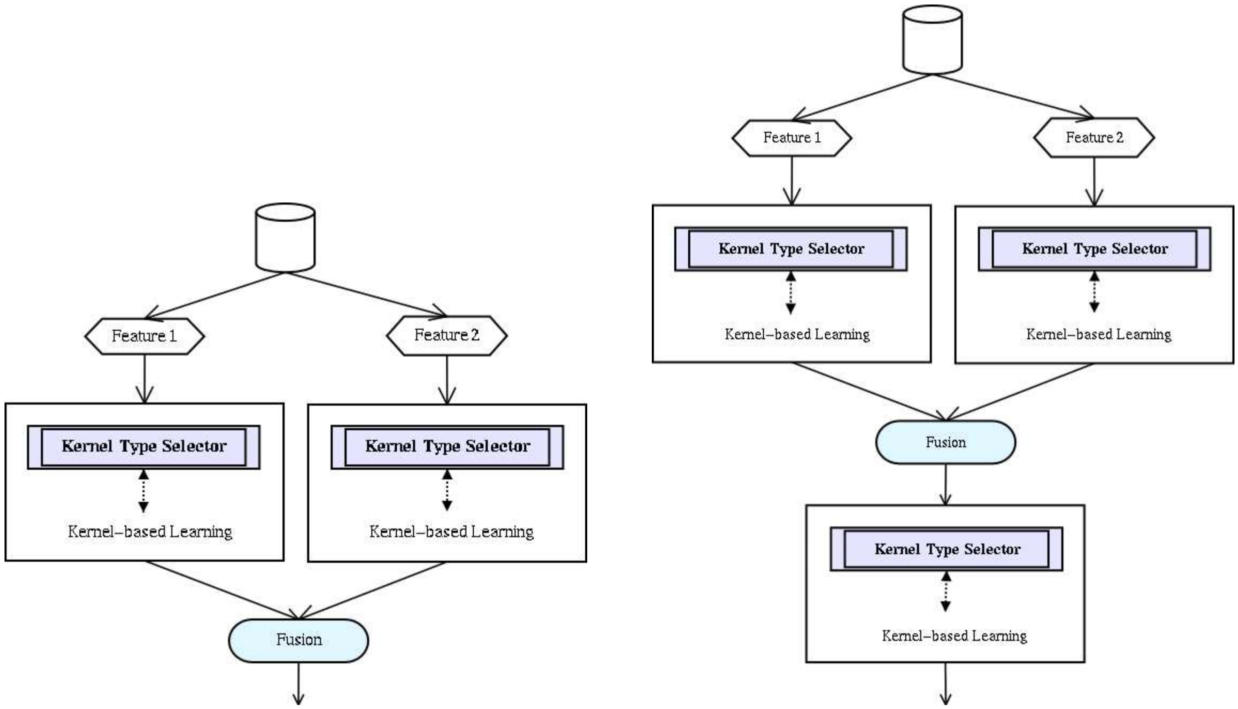
Previous work[9] done with color only revealed that there is no overall best kernel, but maybe there is a best kernel for each query or group of queries. Similarly, here, each feature type might have its own best kernel corresponding to a given query or not. The process is depicted in Figure 1 a) and requires one level of learning.

Many Image Retrieval Systems allow user to introduce the weights α as measures of their importance to the different features.[14,15] A problem that arises is that the user can not easily express his or her queries appropriately in terms of the relative importance of the given feature types (like color, shape). As an example, suppose that a user wants to find images with red circles in blue background. In this case, it is not clear how to assign importance to shape and color features, i.e which of these features is more important for the user.

For this later case, the weight α must be somehow automatically find by the system. We believe that the best way is to obtain these weights dynamically via relevance feedback given by user. As in the previous case, learning can be performed on each feature space separately, and then, the results can be combined. We found,[6] similar to,[4,5] that this fusion approach requires two levels of learning, one for learning the weights of the various features within the feature type and one for learning the parameter α. Therefore, there might be two levels where the kernel selector could be used: one for each feature type, and one for learning the weight α needed for combining the results. In this way, user is no longer required to specify a precise weight for each feature type at the query formulation stage. The process is depicted in Figure 1 b). Lately, this combining process of the results obtained from multiple classifiers has been called "late fusion".[4]

4. EXPERIMENTS AND RESULTS

Previous work[9] done with color only revealed interesting results regarding the relationship between the color and the kernel type. For example, a result was that there is no overall best kernel, but maybe there is a best kernel for each query or group of queries. Also experimental results confirmed those of,[12] which mainly says that a



a) with known weight b) with unknown weight

Figure 2. Fusion after learning.

better choice of the kernel type improves the effectiveness of the retrieval results more than choice of the color space.

In this section, we plan to perform experiments to evaluate the effectiveness and the efficiency of using the fusion schema proposed in Sect. 3.1. The goal is to find out whether or not using the concatenated vector will keep the same query/kernel relationships as found for color.[9, 10]

4.1. Experimental Setup

For our experiments, we use four image test collections of sizes 5000 and 10000, which include 10 and 100 relevant images, respectively, for each query image. For convenience, we name these sets as 5000_10, 5000_100, 10000_10 and 10000_100. All image collections are quantized to 256 colors in RGB and 166 colors in HSV. The collections are annotated by using an approach similar to the one proposed in.[16] We use the same set of 10 images as queries (Q_1,Q_2, ..., Q_{10}) for each experiment (query-by-example approach). We assumed a binary relevance model; each image is either relevant or not.

For evaluation purposes, we use the "Test and Control" method. The process of obtaining the training and testing sets is described in detail in.[17] Images in the training and testing set are randomly distributed. The number of relevant images within the training and the testing sets for each query are given elsewhere.[10, 17] We assume that at each feedback step there are 10 images seen by user. We proceed with learning until 300 images from the training set get feedback from user. To evaluate the quality of retrieval, we use the R_{norm} measure.[18]

For each test collection we perform similar experiments, each one corresponding to a different fusion scheme. We perform the experiments for a set of 12 kernels: 6 polynomials and 6 radial basis, with general forms given respectively by Equations (3) and (4), Section 2.3.

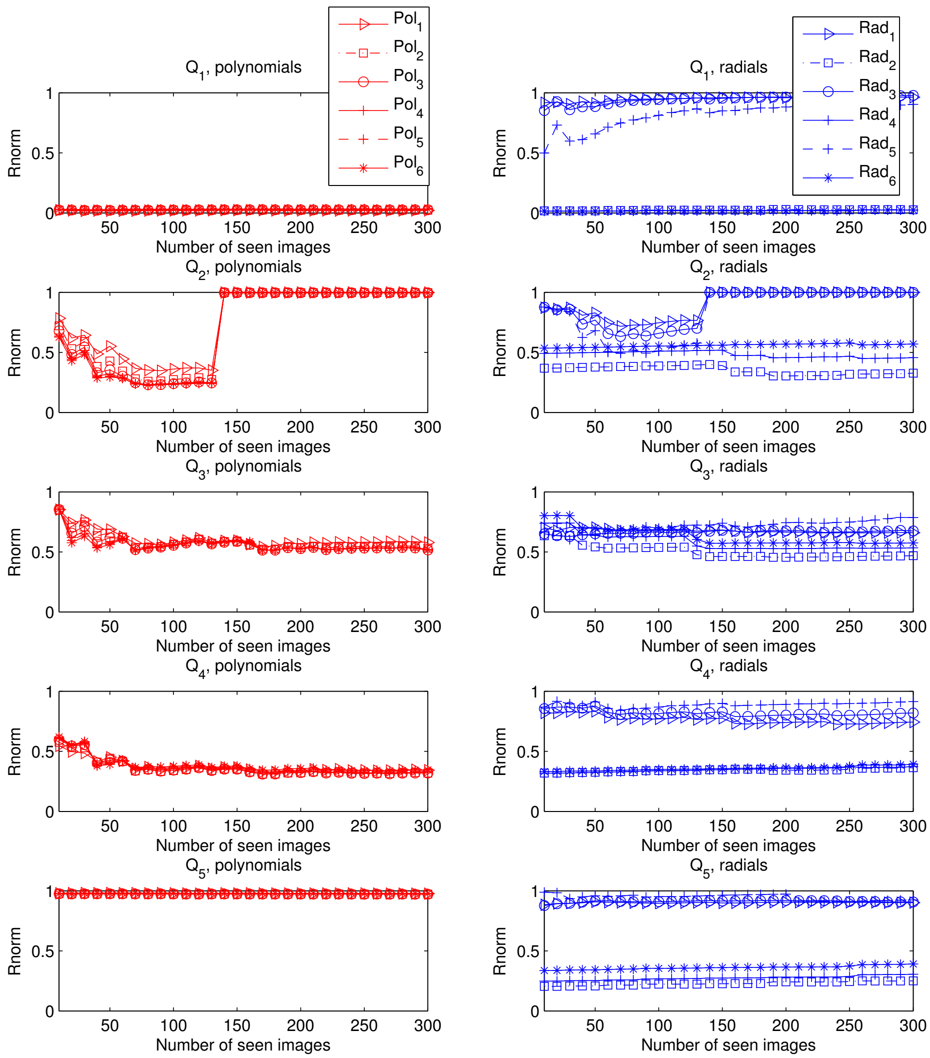


Figure 3. Kernel results for 5000_10 collection for RGB.

4.2. Discussion

Figure 3 shows the results of our experiments for five queries and 5000_10 image collection, with images represented in RGB. However, similar results were obtained for all image collections, for both color spaces.

To summarize the results from our experiments, in Table 2 we give the query groupings according to their best kernel. These groups are calculated for each kernel by using the average R_{norm} values over the feedback steps. The kernel with the highest value is selected as the best kernel for a certain query only if the difference between the respective average value and the next highest average value is more than 0.05 or (less than 0.05) the later corresponds to a less efficient kernel (see[10] for details).

	Pol_1	Pol_2	Pol_3	Pol_4	Rad_1	Rad_3	Rad_5	Rad_6
5000_10	Q_5	Q_7		Q_{10}	Q_1, Q_2, Q_3, Q_8, Q_9	Q_6	Q_4	
5000_100	$Q_2, Q_3, Q_4, Q_6, Q_8, Q_9, Q_{10}$	Q_7			Q_1, Q_5			
10000_10	Q_6, Q_7, Q_8, Q_9		Q_{10}		Q_1, Q_2, Q_5		Q_4	Q_3
10000_100	$Q_3, Q_4, Q_6, Q_7, Q_8, Q_{10}$				Q_5, Q_9	Q_1	Q_2	

Table 2. Query groupings.

As one can notice from Table 2: the more relevant information in the collection, the less complex the kernel type. That is, for collections 5000_100 and 10000_100, there are more queries that have as best kernel a polynomial kernel than for the other two collections. For these, a more complex kernel is chosen as best kernel for most of the queries. Then, the results seem to confirm our results obtained[9, 10] in our previous work.

Next, we want to compare these query groupings with the ones obtained previously,[9, 10, 19] when images were represented only by their color histograms in RGB color space. We give these groupings in Table 3.

	Pol_1	Pol_2	Pol_3	Pol_4	Pol_5	Pol_6	Rad_3	Rad_5	Rad_6
5000_10				Q_8	Q_9	Q_2, Q_3	Q_1, Q_5	Q_4, Q_6, Q_7, Q_{10}	
5000_100	$Q_2, Q_3, Q_4, Q_6, Q_8, Q_9, Q_{10}$						Q_1, Q_5, Q_7		
10000_10	Q_9	Q_8	Q_3, Q_6, Q_7				Q_1, Q_2, Q_4	Q_{10}	Q_5
10000_100	$Q_3, Q_4, Q_6, Q_8, Q_9, Q_{10}$		Q_2				Q_1, Q_5, Q_7		

Table 3. Query groupings with color histograms in RGB.

By comparing the results from Tables 2 and 3 one can notice a strong relationship between the image feature representations (for images and queries) and kernel types. That is, a good feature representation (for images from collection and for the query) can improve the search, whereas a bad one can degrade it. In our experiments, adding annotations was beneficial for most of the cases. For example, for 5000_10 collection for query Q_5 a radial (Rad_3) kernel was the best when only color histograms were used to represent images, and a polynomial (Pol_1) one was the best when both annotations and color histograms were used. Moreover, in the latter case the results (R_{norm} values) are higher than in the first case. That is, for this case, adding annotations improved the system's retrieval performance given by both better results at a faster speed (Pol_1 is more efficient than Rad_3).

In some cases, it seems that using just color histograms may suffice. That is, adding annotations did not change the kernel type. For example, for 5000_100 collection for query Q_3 a polynomial (Pol_1) kernel was the best in both cases. However, there are cases, when it seems that annotations were more confusing than beneficial for the system. A such case is query Q_9 for 10000_100 collection, for which the complexity of the kernel increased from a polynomial (Pol_1) to a radial (Rad_1). As a conclusion, there is a strong relationship between the features representing images and queries, and the kernel type needed to obtain good retrieval results fast.

Next, in Figure 4 we present the results for each collection as averages over the ten queries. These plots show the average performance of each kernel type for a certain collection, and can be used as a guidance in selecting

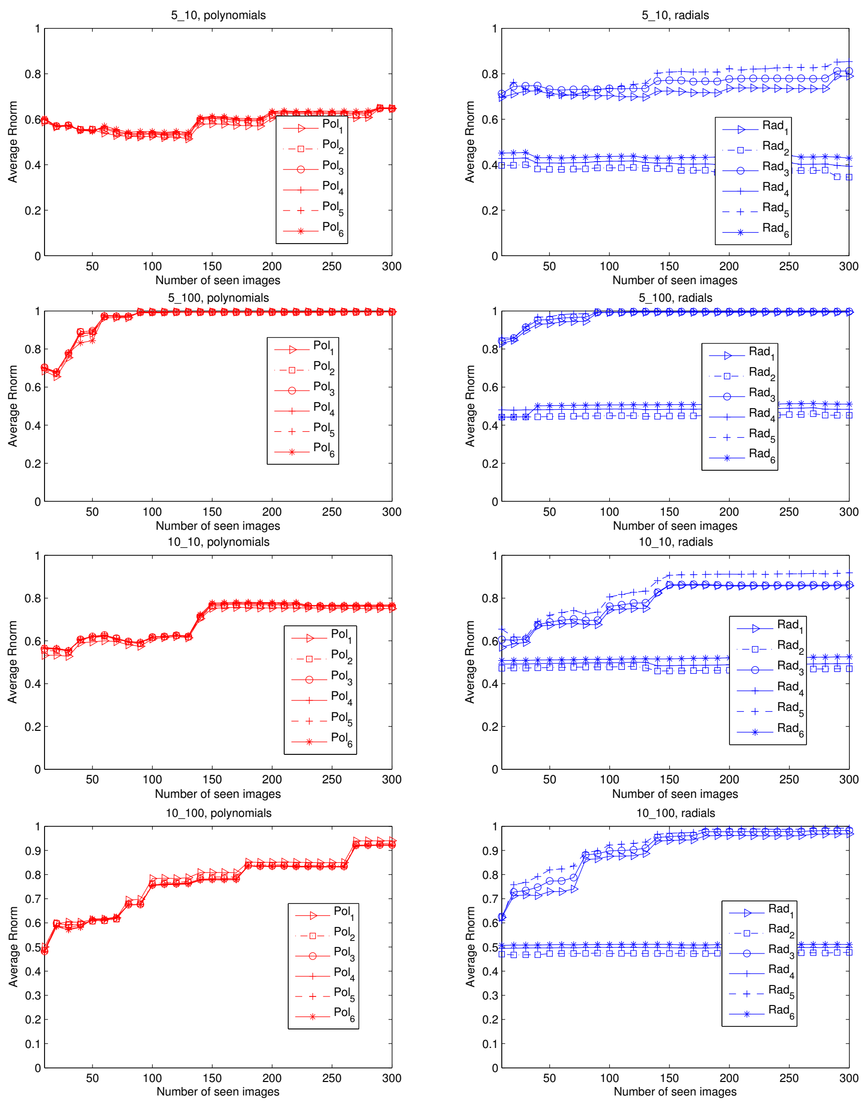


Figure 4. Average results over ten queries for all collections for RGB.

a best kernel for a collection. For example, for all collections, radial kernels Rad_2, Rad_4, and Rad_6 show a low performance, but the other kernels could be a good choice.

In conclusion, the results of these experiments confirm our previous findings obtained for color. That, is:

- there is no general best kernel
- there may be a best kernel for each query or group of queries
- selecting a good kernel is important
- the kernel depends on the feature representations of images and queries

Current work deals with the performance issues raised by the applicability of the score distribution method, proposed in[10] for color only, to images represented by multiple feature types.

5. CONCLUSIONS AND FUTURE WORK

Our previous work on kernel selection issue, performed when color is the only feature representing images, motivates us to further investigate this issue when more than color feature is used to represent images. For this, we incorporate fusion schemes together with kernel selector components into the retrieval system. Several kernels having polynomial and Gaussian Radial Basis Function (RBF) like forms (6 polynomials and 6 RBFs) are applied to generic images represented by annotations and by color histograms in RGB and HSV color spaces.

We implement and test these kernels on four image collections of sizes 5000 and 10000. The experimental results show that an appropriate kernel could significantly improve the performance of the retrieval system. There is a strong relationship between the feature representations of the images and queries, the complexity of the query and the kernel type. As in our previous work, we found that there is no general best kernel, but there may be a best kernel for each query or group of queries. Also, the experiments show that good feature representations can bring in sufficient information such that some queries become easier for the system, which in turn, will require a less complex (possible linear) kernel.

We plan to extend this work to investigate the usage of the kernel types with the fusion schemes presented in Section 3.2. Our future goal is to answer to the following question: Which one of the above fusion schemes is the most effective and efficient to be used for a Web-based Adaptive Image Retrieval System?

REFERENCES

1. W. B. Croft, ed., *Advances in Information Retrieval*, ch. Combining Approaches to Information Retrieval, pp. 1–36. Kluwer Academic Publishers, 2000.
2. R. Fagin, R. Kumar, and D. Sivakumar, "Efficient Similarity Search and Classification via Rank Aggregation," in *Proceedings of the 2003 ACM International Conference on Management of Data, SIGMOD*, pp. 301–312, ACM Press, (San Diego, CA), June 9-12 2003.
3. P. Ciaccia, M. Patella, and P. Zezula, "Processing Complex Similarity Queries with Distance-based Access Methods," in *Proceedings of the 6th International Conference on Extending Database Technology, EDBT, LNCS* **1377**, pp. 9–23, Springer-Verlag, (Valencia, Spain), March 1998.
4. C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, "Early versus Late Fusion in Semantic Video Analysis," in *Proceedings of ACM Multimedia, MM'05*, pp. 399–402, (Singapore), November 6-11 2005.
5. Y. Wu, E. Y. Chang, K. C.-C. Chang, and J. R. Smith, "Optimal Multimodal Fusion for Multimedia Data Analysis," in *Proceedings of ACM Multimedia, MM'04*, pp. 572–579, (New York), October 10-16 2004.
6. A. Doloc-Mihu, V. V. Raghavan, and P. Bollmann-Sdorra, "Color retrieval in vector space model," in *Proceedings of the 26th International ACM SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval MF/IR*, ACM, ed., pp. 1–15, 2003.
7. V. V. Raghavan and S. K. M. Wong, "A Critical Analysis of Vector Space Model for Information Retrieval," *Journal of the American Society for Information Science* **37**(5), pp. 279–287, 1986.

8. M. E. Tipping, "The Relevance Vector Machine," in *Advances in Neural Information Processing Systems, NIPS*, M. S. Kearns, S. A. Solla, and D. A. Cohn, eds., **12**, pp. 652–658, MIT Press, (Cambridge, MA), 2000.
9. A. Doloc-Mihu and V. V. Raghavan, "Selecting the Kernel Type for a Web-based Adaptive Image Retrieval System (AIRS)," in *Internet Imaging VII, Proceedings of SPIE-IS&T Electronic Imaging, SPIE*, **6061**, (San Jose, USA), 18-19 January 2006.
10. A. Doloc-Mihu and V. V. Raghavan, "Using Score Distribution Models to Select the Kernel Type for a Web-based Adaptive Image Retrieval System (AIRS)," in *Proceedings of the 5th ACM International Conference on Image and Video Retrieval (CIVR 2006)*, *LNCS* **4071**, pp. 443–452, Springer-Verlag, (Tempe, Az, USA), 13-15 July 2006.
11. A. Doloc-Mihu, V. V. Raghavan, and P. Bollmann-Sdorra, "Integration of multiple feature types with adaptive retrieval in vector space model," in *Proceedings of the 27th International ACM SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval MF/IR*, ACM, ed., pp. 1–15, 2004.
12. O. Chapelle, P. Haffner, and V. Vapnik, "SVMs for Histogram-Based Image Classification," *IEEE Transactions on Neural Networks* , pp. 1055–1064, 1999.
13. B. V. Dasarathy, "Information/Decision Fusion - Principles and Paradigms," in *Proceedings of the Workshop on Foundations of Information/Decision Fusion with Applications to Engineering Problems*, N. Rao, V. Protopopescu, J. Barhen, and G. Seetharaman, eds., pp. 46–60, (Washington, D.C.), August 7-9 1996.
14. Y. Rui, T. S. Huang, and S. F. Chang, "Image Retrieval: Current Techniques, Promising Directions and Open Issues," *Journal of Visual Communication and Image Representation* **10**(1), pp. 39–62, 1999.
15. R. C. Veltkamp and M. Tanase, "Content-Based Image Retrieval Systems: A Survey," tech. rep., Ultrecht University, 2000.
16. A. B. Benitez and S.-F. Chang, "Perceptual Knowledge Construction from Annotated Image Collections," in *Proceedings of the International Conference on Multimedia & Expo, ICME*, (Lausanne, Switzerland), August 26-29 2002.
17. A. Doloc-Mihu, *Adaptive Image Retrieval System: Retrieval, Learning, Fusion and Visualization.* PhD thesis, University of Louisiana at Lafayette, May 2007. (expected date).
18. P. Bollmann-Sdorra, F. Jochum, U. Reiner, V. Weissmann, and H. Zuse, "The LIVE Project - Retrieval Experiments Based on Evaluation Viewpoints," in *Proceedings of the 8th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 213–214, (Montreal, Canada), June 1985.
19. A. Doloc-Mihu and V. V. Raghavan, "Score Distribution Approach to Automatic Kernel Selection for Image Retrieval Systems," in *Proceedings of the 16th International Symposium on Methodologies for Intelligent Systems (ISMIS 2006)*, *LNAI/LNCS* **4203**, pp. 443–452, Springer-Verlag, (Bari, Italy), 27-29 September 2006.