Using Score Distribution Models to Select the Kernel Type for a Web-based Adaptive Image Retrieval System (AIRS)

Anca Doloc-Mihu and Vijay V. Raghavan

University of Louisiana at Lafayette, Lafayette, LA 70504, USA,
anca@louisiana.edu, raghavan@@cacs.louisiana.edu,
http://www.cacs.louisiana.edu/~axd9917

Abstract. The goal of this paper is to investigate the selection of the kernel for a Web-based AIRS. Using the Kernel Rocchio learning method, several kernels having polynomial and Gaussian forms are applied to general images represented by color histograms in RGB and HSV color spaces. Experimental results on these collections show that performance varies significantly between different kernel types and that choosing an appropriate kernel is important. Then, based on these results, we propose a method for selecting the kernel type that uses the score distribution models. Experimental results on our data show that the proposed method is effective for our system.

1 Introduction

In a Web-based Image Retrieval System, the goal is to answer as well (fast and accurate) as possible with images that meet the user's request. Here, we assume that the database and the query image(s) are represented by color histograms. However, a characteristic of these colors is that they are not independent, but correlated. Moreover, the interaction between colors is stronger for some queries (images) than for other queries. In the former case, a more complex, possibly non-linear, kernel has to be used, whereas in the latter case, a linear kernel may be sufficient. 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. However, in the literature, "there are currently no techniques available to learn the form of the kernel" [4]. Methods like Relevance Vector Machines [11] assume distribution of data, which might fit or not a real collection. In this study, we learn from real tests the characteristics of our data, and then, we build our kernel selection method.

The paper organization is as follows. Section 2 provides the retrieval model, and the kernel method. Based on the experimental results presented in section 3, section 4 explores a new methodology for selecting the kernel to be used by the Kernel Rocchio learning method in order to improve the performance of an adaptive image retrieval system (AIRS). Finally, section 5 concludes the paper.

2 Background

2.1 Retrieval Model

In Image Retrieval, the user searches a large collection of images to find images that are similar to a specified query. The search is based on the similarities of the image attributes (or features) such as colors. A linear retrieval form [2, 1] matches image queries against the images from collection

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

The query image Q contains the features desired by user. The bigger the value of the function F, when applied to a query Q and an image P, the better is the match between the query image Q and the database image P. For representing images by color, we use the histogram representations in RGB and HSV color spaces.

2.2 Kernel Method

The kernel method constitutes a very powerful tool for the construction of a learning algorithm by providing a way for obtaining non-linear decision boundaries from algorithms previously restricted to handling only linearly separable datasets.

In this work, the general form of the polynomial kernels [4, 1] is given by

$$K(\mathrm{x}, \mathrm{y}) = \left(\langle \mathrm{x}, \mathrm{y} \rangle\right)^{\mathrm{d}}, \mathrm{d} > 0, \tag{1}$$

and the general form of the radial kernels is given by

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

Recently, image retrieval systems started using different learning methods for improving the retrieval results. In this work, we use the **Kernel Rocchio** method [1] for learning.

3 Experimental Study of Kernel Type Selection

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. Therefore, in this section, we experimentally study, through several test query examples, the possible relationships between the different queries and the different kernels.

3.1 Experimental Setup

For our experiments, we use two test collections of size 5000, which include 10 and 100 relevant images, respectively, for each query image, and one of size 10000, which includes 100 relevant images for each query image. For convenience, we name these sets as 5000_10, 5000_100 and 10000_100. All image collections are quantized to 256 colors in RGB and 166 colors in HSV. We use the same set of 10 images as queries (Q_1,Q_2, ..., Q_{10}) for each experiment.

For evaluation purposes, we use the Test and Control method. The process of obtaining the training and testing sets is described in detail in [6]. For each test collection we create 3 different training sets, each of 300 images (called Set 1, Set 2, Set 3), and one test set. The images in the three training sets and the testing set are randomly distributed. The number of relevant images within the training and the testing sets for each query is given in Table 1. We assume that at each feedback step there are 10 images seen by user. For each test collection (5000_10, 5000_100 and 10000_100), we perform three similar experiments, each one corresponding to a different training set (Set 1, Set 2, Set 3). That is, in total, we performed 9 experiments.

Table 1. Number of the relevant images within the training sets and the testing set.

5000_10	Set 1	Set 2	Set 3	Test
Q_1	0	0	2	4
Q_2	1	0	0	4
Q_3	0	1	0	7
Q_4	0	0	1	7
Q_5	1	0	0	7
Q_6	0	0	2	4
Q_7	0	1	0	5
Q_8	1	1	0	4
Q_9	0	0	0	2
Q_{10}	1	1	0	4

5000_100	Set 1	Set 2	Set 3	Test
Q_1	9	8	4	53
Q_2	6	5	5	50
Q_3	4	8	9	51
Q_4	8	5	2	56
Q_5	7	8	6	43
Q_6	5	9	1	56
Q_7	4	7	9	56
Q_8	4	5	7	48
Q_9	5	5	3	48
Q_{10}	6	8	9	50

10000_100	Set 1	Set 2	Set 3	Test
Q_1	1	4	2	62
Q_2	0	5	4	49
Q_3	4	3	2	52
Q_4	7	1	5	46
Q_5	7	3	3	47
Q_6	7	5	3	46
Q_7	3	1	4	44
Q_8	4	4	5	46
Q_9	1	3	1	49
Q_{10}	2	2	2	48

To evaluate the quality of retrieval, we use the R_{norm} measure [3]. We perform the experiments for a set of 12 kernels: 6 polynomials and 6 radial basis, with general forms given respectively by Equations (1) and (2). The values of the parameters (a, b, c, and d) and the names of the kernels used in the experiments are presented in Table 2. In all experiments presented in this work, $\sigma = 1$.

Space limitation precludes us from showing all the plots obtained from our experiments. However, as an example, in Figure 1, we present the plots of the kernel values obtained for query Q_1 for 5000_100 image test collection in RGB color space, for the first training set, Set 1.

3.2 Discussion of the Results

In this section, we analyze the possible relationships between the different queries and the different kernels that occur in our experiments.

Table 2. 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

Selecting the kernel for a given query. In our previous work [6], we found that there is no general best kernel, but there may be a best kernel for each query or groups of queries. Continuing this work, from Figure 1 we notice the different behaviors of the different kernels for query Q_1. Moreover, each query within each of the 9 experiments presents similar behavior. To be able to study the results of our experiments, we need to define our criterion for selecting the best kernel corresponding to a particular query.

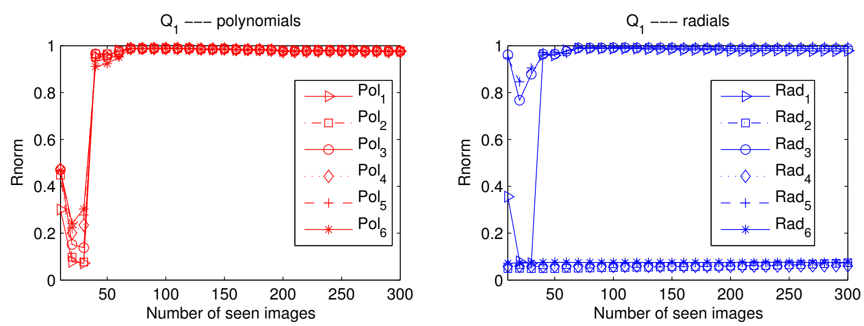

Fig. 1. Kernel results for query Q_1 for 5000_100 in RGB.

For this, for each query within each training set of each test collection, we compute the average R_{norm} (over all feedback steps) value corresponding to each kernel, and then, we compare these 12 values. The kernel with the highest average R_{norm} value is chosen as the best initial kernel for the respective query. If for any query there is a kernel more efficient than the initial kernel, whose average R_{norm} value is very close to the initial best kernel (the difference between the two values is less than 0.05), then this kernel is chosen as the best final kernel for that query. The order of our kernels, from the more efficient to the less efficient, is Pol_1, ..., Pol_6, for polynomials and Rad_2,Rad_1,Rad_4,Rad_6,Rad_3,Rad_5 for radials. Then, low order polynomials (Pol_1,Pol_2) are always preferred over any radials.

Table 3 presents this process of choosing the best kernel for all 10 queries for 5000_100 collection for the first training set (Set 1) in RGB color space. By using this procedure, we select the best (final) kernels for all runs for all 3 test collections in both color spaces, which we use in our study from now on.

Table 3. Selecting the best (final) kernel for 5000_100 test collection, for Set 1 in RGB.

RGB	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}
Initial	Rad_5	Pol_5	Rad_5	Rad_3	Rad_5	Rad_5	Rad_5	Pol_5	Rad_5	Rad_5
Difference	0.009	0.005	0.007	0.014	0.007	0.025	0.033	0.007	0.006	0.0184
Final	Rad_3	Pol_1	Pol_1	Pol_1	Rad_3	Pol_1	Rad_3	Pol_1	Pol_1	Pol_1

Query groupings based on kernel type. To summarize the results from our experiments, in here, we start by grouping the queries according to their performance with respect to the different best kernels, for each experiment. These groupings are shown in Tables 4 and 5, for 5000_100 and 10000_100 test collections, respectively.

Table 4. Query groupings for 5000_100 test collection.

Set 1	Rad_2	Rad_3	Pol_1	Pol_2	Pol_3
RGB		Q_1, Q_5, Q_7	$Q_2, Q_3, Q_4, Q_6, Q_8, Q_9, Q_{10}$		
HSV	Q_3, Q_{10}		$Q_2, Q_4, Q_5, Q_7, Q_8, Q_9$	Q_6	Q_1

Set 2	Rad_1	Rad_2	Rad_3	Rad_5	Pol_1	Pol_2
RGB			Q_5		$Q_1, Q_2, Q_3, Q_4, Q_6, Q_7, Q_8, Q_9$	Q_{10}
HSV	Q_1	Q_9		Q_4	$Q_2, Q_3, Q_5, Q_6, Q_7, Q_8, Q_{10}$	

Set 3	Pol_1	Pol_2	Pol_6
RGB	$Q_2, Q_3, Q_4, Q_5, Q_6, Q_7, Q_8, Q_9$	Q_{10}	Q_1
HSV	$Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7, Q_8, Q_9, Q_{10}$		

Kernels' behavior in different color spaces. It is known that HSV color space is more attractive, due to its approximately perceptually uniform characteristic, than the RGB color space. In [4] the authors found that, in practice, "the impact of the choice of the color space on performance" is minimal when "compared to the impacts of the other experimental conditions" such as the choice of the kernel type. An explanation is that the classifier does not use any information about the color space after quantization [4]. By using the results presented in the previous tables, we group the queries according to which color space has a more efficient best kernel type in Table 6. As we can see from the table, there are some queries that perform equally well (using the same efficient kernel) in both color spaces (column marked "Same"). Then, there are cases where it seems that it is better to work in RGB color space than in HSV color

Table 5. Query groupings for 10000_100 test collection.

Set 1	Rad_2	Rad_3	Rad_5	Pol_1	Pol_3	Pol_5
RGB		Q_1, Q_5, Q_7		$Q_3, Q_4, Q_6, Q_8, Q_9, Q_{10}$	Q_2	
HSV	Q_2		Q_1	$Q_3, Q_4, Q_5, Q_6, Q_8, Q_9, Q_{10}$		Q_7

Set 2	Rad_3	Rad_5	Pol_1	Pol_2	Pol_3
RGB	Q_1, Q_7		Q_2, Q_5, Q_8, Q_9	Q_3, Q_4, Q_6	Q_{10}
HSV		Q_1	$Q_2, Q_3, Q_5, Q_8, Q_9, Q_{10}$		Q_4, Q_6, Q_7

Set 3	Rad_3	Rad_5	Pol_1	Pol_2	Pol_3
RGB	Q_5	Q_7	$Q_1, Q_2, Q_4, Q_6, Q_8, Q_9, Q_{10}$	Q_3	
HSV	Q_7		$Q_2, Q_3, Q_4, Q_6, Q_8, Q_9, Q_{10}$	Q_1	Q_5

space (e.g., for the Set 2, for the 10000_100 collection, there are 5 queries in RGB and 3 in HSV), and vice-versa. This result is consistent with the results of [4].

Table 6. Query grouping according to the space representation for Set 2.

Set 2	RGB	Same	HSV
5000_10	Q_1, Q_2, Q_7, Q_9	Q_8	$Q_3, Q_4, Q_5, Q_6, Q_{10}$
5000_100	Q_1, Q_4, Q_9	Q_2, Q_3, Q_6, Q_7, Q_8	Q_5, Q_{10}
10000_100	Q_1, Q_4, Q_6, Q_8, Q_9	Q_2, Q_5	Q_3, Q_7, Q_{10}

Kernels' behavior across different collections. In this part, we want to answer the question of whether a query keeps the same best kernel across different test collections or not. From Tables 4 and 5, one can notice that there are queries that keep the same best kernel across both test collections, but only for some training set(s), and not for all 3 training sets of each collection. For example, in RGB, query Q_1 for Set 1 gets Rad_3 as the best kernel across both test collections, and for Set 2, it gets Pol_1 and Rad_3 as the best kernels for 5000_100 and 10000_100 test collections, respectively. As a conclusion, in our experiments, no particular query has a best kernel across different collections.

Kernels' behavior for different training sets. For this study, we perform 3 runs, each corresponding to a different training set (Set 1, Set 2, Set 3), on each test collection. From Tables 4 and 5, one can notice that there are some queries that have the same kernel for all 3 training sets. The other queries, in either color space, have different kernels between the 3 training sets, without a general pattern. In conclusion, since different runs (training sets) include different number of relevant images and cases differ in terms of if they offer enough information or not, the choice of the best kernel between the different training sets depends on how much information (from relevant and non-relevant images) is good enough for the respective query. If this information is equally good between the runs then they show the same best kernel, and conversely.

Influence of the number of relevant images. In here, we analyze the effect of having different number of relevant images in our test collections, or generality [5]. As a statistics: for the 5000_10 test collection, with a generality of 0.002, radial kernels represent 50% in RGB color space, and 40% in HSV; for the 5000_100 test collection, with a generality of 0.02, radial kernels represent 13% in RGB color space, and 17% in HSV; for the 10000_100 test collection, with a generality of 0.01, radial kernels represent 23% in RGB color space, and 13% in HSV. That is, in RGB color space, the smaller the generality fraction, the bigger is the number of queries that present a radial kernel as their best kernel. However, it seems that in HSV color space this rule does not necessarily apply.

From our plots, one can notice that, generally, all polynomial kernels and Rad_3, Rad_5, Rad_1 kernels display the same increasing curves (and behavior) whenever a relevant image is learned. On the other side, the other three radial kernels display, in general, similar behavior, approximately constant. However, there are cases when Rad_2 kernel's curve drops when a relevant image is learned, and cases when Rad_4, Rad_6 curves increase when a relevant image is learned. As a statistics, Pol_1 is the best kernel for the most number of queries (55%), Rad_3 is chosen by 13% of queries, Pol_2 is chosen by 7% of queries, Pol_3 is chosen by 8% of queries, Rad_5 is chosen by 6% of queries, whereas the other kernels are chosen by less than 5% of the queries. From these observations, we can prune down the number of kernels to study from 12 to 5 (Pol_1,Pol_2,Pol_3,Rad_3,Rad_5), which answer to approximately 90% of our queries.

4 Using Score Distributions for Kernel Type Selection

Researchers [8, 9, 7] modeled the score distributions of search engines for relevant and non-relevant documents by using a normal and an exponential distribution, respectively. In this section, we propose a procedure based on this model to select the kernel type for an AIRS.

4.1 Mathematical Model of Image Score Distributions

The score distributions of the relevant documents suggested by researchers [9] are modeled as

$$P(score|R = r) = \frac{1}{\sqrt{2\pi\sigma}} \exp(-\frac{(score-\mu)^2}{2\sigma^2}),$$

where μ is the mean, and σ is the variance of the Gaussian distribution. The score distributions of the non-relevant documents are modeled as

$$P(score|R = nr) = \lambda \exp(-\lambda * score),$$

where λ is the mean for the exponential distribution.

Our experimental results (Section 3.1) support these mathematical models of the score distributions, with some approximation. This analysis was done for 5000_100 and 10000_100 test collections only, since the 5000_10 test collection does not contain sufficient number of relevant images and, therefore, the score distribution models do not fit for this collection. Figure 2 illustrates how the score distributions fit the top 300 images for one query for 3 kernels.

As an observation, the score distributions are different for the different kernel types used in the experiments. This motivates us to seek for a method to select the kernel type by using these differences between the score distributions. Next, we present such a method.

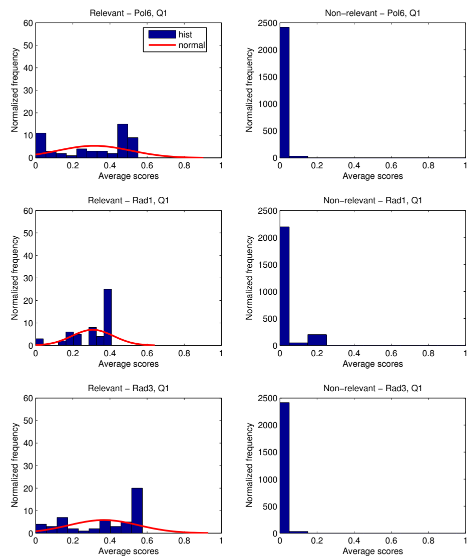

Fig. 2. Score distributions for query Q_1 for 5000_100 and Set 1 in RGB, for Pol_6, Rad_1, Rad_3 kernels.

4.2 Kernel Selection Method Based on Image Score Distributions

In image retrieval, the goal is to retrieve as many relevant images and as few non-relevant images as possible. This means, we wish to have a retrieval system capable to rank all the relevant images before the non-relevant ones. In other words, the scores of the relevant images should be as high as possible, whereas the scores of the non-relevant images should be as low as possible.

In our system, these scores are computed by using the different kernels type. Then, logically, the best kernel is the one that is able to distribute the scores of the images such that the relevant images have higher scores than the non-relevant ones. Therefore, we suggest the following procedure for selecting the kernel type:

1. for each kernel type K_i, compute
   (a) the score distribution of the relevant images (ScR_i) by using a Gaussian, and
   (b) the score distribution of the non-relevant images ($ScNR_i$) by using an exponential.
2. compare the results of any two kernel types K_i and K_j; reject those kernels K_j for which there is a kernel K_i with $\sum_{s=1}^{10} ScR_{is} > \sum_{s=1}^{10} ScR_{js}$ and $\sum_{s=1}^{10} ScNR_{is} < \sum_{s=1}^{10} ScNR_{js}$, where $ScR_{is} = NormalizedFreq_{is} * score_{is}$ and s is the score interval.

As an observation, the above procedure can be applied after each feedback step to select the best kernel type to be used for the following feedback step. Another observation is that the model is biased, especially for low scoring images, which do not occur between the top 300 images. However, for high scoring images, the model offers a relatively good estimation [8].

Discussion. Our method of selecting the best kernel tries to fit the score distributions of both relevant and non-relevant images, such that there are as many as possible relevant images with high scores grouped towards the right half of the plot, and less relevant images grouped in the left half side, and vice-versa for the non-relevant images.

For example, in Figure 2, Rad_3 and Pol_6 have better distributions for the non-relevant images than Rad_1 (more non-relevant images get scores of 0 or below). For relevant images, Rad_1, followed by Pol_6, shows the least number of relevant images with scores of 0. But, Rad_3, followed by Pol_6, shows a bigger number of relevant images with bigger scores. As a result, by cumulating these observations, the order for the best kernel is Rad_3, Pol_6, and Rad_1.

If we compare this result with the results from the previous section 3, we can see that these kernels display very close results (Figure 1). However, this result is consistent with our result from the experiments (Table 3). We obtained similar results for most of the queries in all the experiments.

As a conclusion, our method for selecting the kernel type for a particular query is based on score distributions, which are obtained via feedback from user and can be calculated automatically by the system. The method gives the same results as those obtained in Section 3, from extensive experiments, for most of the cases. That is, this method could be a viable solution to automatically select the kernel type in an AIRS.

5 Conclusions and Future Work

Kernel methods offer an elegant solution to increase the computational power of the linear learning algorithms by facilitating learning indirectly in high - dimensional feature spaces. Therefore, kernels are important components that can improve the retrieval system.

This motivates us to investigate several types of kernels in order to improve the performance as well as the response time of our AIRS, which is intended to be a web-based image retrieval application. For this, several kernels having polynomial and Gaussian Radial Basis Function (RBF) like forms (6 polynomials and 6 RBFs) are applied to generic images represented by color histograms in RGB and HSV color spaces.

We implement and test these kernels on image collections of sizes 5000 and 10000. Experimental results on these collections show that an appropriate kernel could significantly improve the system performance. By observing the behavior of the different kernels for several queries, we answer to several questions about the possible characteristics that might influence the kernels' behavior. Then, based on these observations, we propose a kernel selection method that uses score distribution models to select the best kernel for a particular query. The method shows approximately the same results in selecting the best kernel type for most of the queries and parameter settings used in our experiments.

As future work, we plan to investigate whether our kernel selection method works or not when multiple feature types (e.g., color and texture) are used to represent images.

References

1. Doloc-Mihu, A., Raghavan, V., V., Bollmann-Sdorra, P.: Color Retrieval in Vector Space Model. Proceedings of the 26th International ACM SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval, MF/IR (2003)
2. Raghavan, V., V., Wong, S. K. M.: A Critical Analysis of Vector Space Model for Information Retrieval. Journal of the American Society for Information Science **37** (1986) 279-287
3. Bollmann-Sdorra, P., Jochum, F., Reiner, U., Weissmann, V., Zuse, H.: The LIVE Project - Retrieval Experiments Based on Evaluation Viewpoints. Proceedings of the 8th International ACM SIGIR Conference on Research and Development in Information Retrieval (1985) 213–214
4. Chapelle, O., Haffner, P., Vapnik, V.: SVMs for Histogram-Based Image Classification. IEEE Transactions on Neural Networks **5** (1999) 1055–1064
5. Huijsmans, D. P, Sebe, N.: How to Complete Performance Graphs in Content-Based Image Retrieval: Add Generality and Normalize Scope. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 245–251
6. Doloc-Mihu, A., Raghavan, V., V.: Selecting the Kernel Type for a Web-based Adaptive Image Retrieval System (AIRS). Internet Imaging VII, Proceedings of SPIE-IS&T Electronic Imaging, SPIE **6061** (2006) 60610H
7. Swets, J., A.: Information Retrieval Systems. Science **141** (1963) 245-250
8. Manmatha, R., Feng, F., Rath, T.: Using Models of Score Distributions in Information Retrieval. Proceedings of the 24th ACM SIGIR Conference on Research and Development in Information Retrieval (2001) 267-275
9. Arampatzis, A., Beney, J., Koster, C., van der Weide, T., P.: Incrementally, Half-Life, and Threshold Optimization for Adaptive Document Filtering. The 9th Text REtrieval Conference (TREC-9) (2000)
10. Cristianini, N., Shawe-Taylor, J.: An introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press (2000)
11. Tipping, M., E.: The Relevance Vector Machine. Advances in Neural Information Processing Systems **12** (2000) 652-658