Feature Extraction Using Low-Rank Approximations of the Kernel Matrix

A.R. Teixeira[1], A.M. Tomé[1], and E.W. Lang[2]

[1] DETI/IEETA, Universidade de Aveiro, 3810-193 Aveiro, Portugal
ana@ieeta.pt
[2] CIMLG, Institute of Biophysics, University of Regensburg, D-93040 Regensburg, Germany
elmar.lang@biologie.uni-regensburg.de

Abstract. In this work we use kernel subspace techniques to perform feature extraction. The projections of the data onto the coordinates of the high-dimensional space created by the kernel function are called features. The basis vectors to project the data depend on the eigendecomposition of the kernel matrix which might become very high-dimensional in case of a large training set. Nevertheless only the largest eigenvalues and corresponding eigenvectors are used to extract relevant features. In this work, we present low-rank approximations to the kernel matrix based on the Nyström method. Numerical simulations will then be used to demonstrate the Nyström extension method applied to feature extraction and classification. The performance of the presented methods is demonstrated using the USPS data set.

1 Introduction

Kernel techniques are often claimed to have better performance in feature extraction applications because the non-linear structure of the data is retained. Kernel subspace techniques are projective methods in a feature space created by a non-linear transformation $\phi(\mathbf{x})$ of the data. The data is thereby mapped into an high (and possible infinite) dimensional space through a nonlinear transformation. However, the explicit mapping into feature space is avoided by using kernel functions which define a dot product $k(i,j) = \phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j)$ in feature space that can be computed implicitly using data in input space [8]. Then every data manipulation (or every algorithm) can be efficiently computed as long as it can be translated into a sequence of dot products. The kernel matrix $\mathbf{K}$ represents a matrix of dot products of the mapped data and can easily be computed as its entries $k(i,j)$ depend on the corresponding input data points $(\mathbf{x}_i, \mathbf{x}_j)$ and are computed according to the defined kernel function. However, the dimension of the kernel matrix depends on the size of the data set which can become prohibitively large in certain applications.

For such large data sets, the size of the kernel matrix then represents a bottleneck because of the computational burden once its eigendecomposition must be achieved. In such cases projective subspace techniques can be invoked. Their goal is to describe the data set with a subset with reduced dimensionality by extracting meaningful components while still retaining the inherent structure of the original data set. Subspace

A. Campilho and M. Kamel (Eds.): ICIAR 2008, LNCS 5112, pp. 404–412, 2008.

techniques only compute the projections of the data vectors onto the directions corresponding to the most significant eigenvalues of the kernel (or covariance) matrix. To achieve a low rank approximation to a complete eigendecomposition of the kernel matrix, the Nyström extension is a strategy that can be exploited as discussed in [6], [10]. The Nyström extension to the eigenvector matrix is based on the eigendecomposition of sub-blocks of the full kernel matrix. As a consequence, only a subset of the complete training set is used to approximately represent the eigenvectors. Several works [3], [7], [1], [4] related with kernel principal component analysis (KPCA) have suggested a similar strategy.

In this work we consider the Nyström extension computed using the incomplete Cholesky decomposition of the kernel matrix. The method has a very efficient implementation if the kernel is based on radial basis functions. Furthermore, the identification of an appropriate subset of the training data set that forms the basis vectors onto which the data vectors are to be projected is automatically determined by the algorithm. We illustrate the method by extracting relevant features of the USPS data set which can be used in a classification task. The numerical simulations compare the performance of the classifiers using kernel features versus principal component (PCA) features.

2 Low Rank Approximation of Kernel Matrix

Applying kernel methods, an eigendecomposition of the related kernel matrix, and particularly the most significant eigenvalues and corresponding eigenvectors, are often required. For large training data sets, the corresponding kernel matrix $\mathbf{K}$ becomes prohibitively large. Consequently, its eigendecomposition is often impractical in real data applications. In such cases an appropriate dimension reduction must be achieved. Few papers [6], [10] discuss the application of the Nyström extension method to compute a low rank approximation of the kernel matrix $\tilde{\mathbf{K}} = \mathbf{V}\mathbf{D}\mathbf{V}^T$ where only the R largest eigenvalues and corresponding eigenvectors are computed. The method is based on the fact that the kernel matrix can be written in block notation [10], [6] as follows:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_r & \mathbf{K}_{rs} \\ \mathbf{K}_{rs}^T & \mathbf{K}_s \end{bmatrix} \tag{1}$$

Considering that the full matrix has dimension $K \times K$, the upper left block matrix $\mathbf{K}_r$ has dimension $R \times R$, the upper right block matrix $\mathbf{K}_{rs}$ has dimension $R \times (K-N)$ and the lower right block matrix $\mathbf{K}_s$ has dimension $S \times S$ where $S = K - R$. This notation implicates that the mapped training data set of dimension K is divided into two subsets of size R and $S = K - R$, respectively. The matrix $\mathbf{K}_r$ represents the kernel matrix within subset $\mathbf{\Phi}_R$ (with R vectors), $\mathbf{K}_{rs}$ is the kernel matrix comprising subsets $\mathbf{\Phi}_R$ and $\mathbf{\Phi}_S$ and $\mathbf{K}_s$ is the kernel matrix of the subset $\mathbf{\Phi}_S$.

The low-rank approximation is written using the block matrices $\mathbf{K}_r$ and $\mathbf{K}_{rs}$ according to [10], [6]

$$\tilde{\mathbf{K}} = \begin{bmatrix} \mathbf{K}_r \\ \mathbf{K}_{rs}^T \end{bmatrix} \mathbf{K}_r^{-1} \begin{bmatrix} \mathbf{K}_r & \mathbf{K}_{rs} \end{bmatrix} \tag{2}$$

It can be shown that the lower block is approximated by $\mathbf{K}_s \approx \mathbf{K}_{rs}^T \mathbf{K}_r^{-1} \mathbf{K}_{rs}$. The Nyström extensions for the R eigenvectors $\mathbf{V}$ corresponding to the R largest eigenvalues are obtained as

$$\mathbf{V}^T = \mathbf{H}^T \left[\mathbf{K}_r \; \mathbf{K}_{rs} \right] \tag{3}$$

The matrix $\mathbf{H}$ is computed using eigendecompositions of $R \times R$ matrices, where R is the size of subset $\mathbf{\Phi}_R$. Different approaches were considered to form the $R \times R$ matrices: in [10] only the block $\mathbf{K}_r$ is considered while in [6] it is additionally computed a matrix related with both upper blocks of the kernel matrix. The main difference between both approaches is that eigenvectors are non-orthogonal [10] or orthogonal [6].

2.1 Computing the Eigenvectors

In this work we are interested on the solution that leads to orthogonal eigenvectors, $\mathbf{V}^T\mathbf{V} = \mathbf{I}$. In [6] a solution is proposed which uses as starting point the eigendecomposition of the block matrix $\mathbf{K}_r$. The latter is formed by either randomly selecting elements of the training set or rows/columns of $\mathbf{K}$. This result is used to transform the data and compute a new $R \times R$ matrix whose eigendecomposition will also contribute to the eigenvector matrix. Here we instead use the proposal of [2] which is based on the incomplete Cholesky decomposition using a symmetric pivoting scheme. The incomplete Cholesky decomposition leads to

$$\mathbf{C} = \left[\mathbf{L} \;\; \mathbf{L}^{-T} \mathbf{K}_{rs} \right] \tag{4}$$

The matrix $\mathbf{L}$ represents a triangular matrix corresponding to the complete Cholesky decomposition of $\mathbf{K}_r = \mathbf{L}^T\mathbf{L}$. Notice that the identification of the matrix $\mathbf{L}$ arises naturally with the pivoting scheme and does not need to be known in advance. So, the pivoting index of the incomplete Cholesky decomposition [2] leads to the selection of $\mathbf{\Phi}_R$ from the training set.

Considering that the kernel matrix can be approximated by the incomplete Cholesky $\tilde{\mathbf{K}} = \mathbf{C}^T\mathbf{C}$, its low-rank approximation can also be derived from an $R \times R$ matrix defined by

$$\mathbf{Q} = \mathbf{C}\mathbf{C}^T = \mathbf{V}_q \mathbf{D} \mathbf{V}_q^T \tag{5}$$

The result of this eigendecomposition as well as the decomposition of $\mathbf{K}_r$ leads to

$$\mathbf{H} = \mathbf{L}^{-1} \mathbf{V}_q \mathbf{D}^{-1/2} \tag{6}$$

Substituting this result into the eigenvector equation (3) yields

$$\mathbf{V} = \begin{bmatrix} \mathbf{K}_r \\ \mathbf{K}_{rs}^T \end{bmatrix} \mathbf{L}^{-1} \mathbf{V}_q \mathbf{D}^{-1/2} \tag{7}$$

It can be easily shown that the Nyström extension to the eigenvector matrix $\mathbf{V}$ has R orthogonal eigenvectors.

3 The Data in Feature Space and Basis

In the feature space the mapped data set $\mathbf{\Phi}$ is represented by its projections

$$\mathbf{Z} = \mathbf{U}^T \mathbf{\Phi} \tag{8}$$

where the columns of the matrix $\mathbf{U}$ form a basis in feature space onto which the data set is projected. In subspace techniques the basis can always be expressed as a linear combination of the mapped data [8]

$$\mathbf{U} = \mathbf{\Phi}_B \mathbf{A} \tag{9}$$

The matrix $\mathbf{A}$ is a matrix of coefficients and either $\mathbf{\Phi}_B$ is the complete training data set or a subset of the data set only. Furthermore, the projections $\mathbf{Z}$ of the training set are also related with the eigenvectors of the kernel matrix ($\mathbf{K}$) of the data set. Considering a singular value decomposition of the data set

$$\mathbf{\Phi} = \mathbf{U}\mathbf{D}^{1/2}\mathbf{V}^T \tag{10}$$

where $\mathbf{D}$ is a diagonal matrix with its non-zero eigenvalues of the kernel matrix (or of the scatter matrix) ordered according to $\lambda_1 > \lambda_2 > \ldots > \lambda_L \ldots > \lambda_R$ and $\mathbf{V}$ and $\mathbf{U}$ are the R eigenvectors of the kernel and covariance matrices, respectively. Note that the square root of the eigenvalues form the singular values of the data matrix. The data set can be approximated using an SVD decomposition with the R most significant singular values and the corresponding eigenvectors. By the manipulation equations (10), (8) and (3) the projections for each element of the training data set read

$$\mathbf{Z} = \mathbf{D}^{1/2}\mathbf{V}^T = \mathbf{V}_q^T \mathbf{L}^{-T} \left[\mathbf{K}_r \; \mathbf{K}_{rs} \right] = \mathbf{V}_q^T \mathbf{L}^{-T} \mathbf{\Phi}_R^T \left[\mathbf{\Phi}_R \; \mathbf{\Phi}_S \right] \tag{11}$$

Comparing the previous result with eqn.(8), the basis vector matrix can be written as

$$\mathbf{U} = \mathbf{\Phi}_R \mathbf{L}^{-1} \mathbf{V}_q \tag{12}$$

It has to be noticed that the R vectors form an orthogonal basis in the feature space, i.e., $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. The eigenvectors in the matrix $\mathbf{V}_q$ should be placed according to their corresponding eigenvalues. The first column should have the eigenvector corresponding to the largest eigenvalue and so on. Furthermore the matrix can have $L < R$ columns to enable projections of the data onto the directions related to the L largest eigenvalues.

4 Numerical Simulations

The relevant features to be used in classification are the projections onto basis vectors computed in the input space and in the feature space. In the input space, the basis vectors are deduced using principal component analysis (PCA), i.e. they form the eigenvectors of the covariance matrix of the data set [5]. In feature space, the basis vectors are computed as described in the last sections. Anyway, the basis vectors either in input space or in feature space are computed using samples from the training data set. In classification problems, during the training step the projections of the training data set on the basis

vectors are used to train the classifiers. During the recall phase, the projections of the test data set onto the same basis vectors are then used to evaluate the performance of the classifier.

Data set description. The USPS data set (accessible at www.kernel-machines.org) is divided into a training data set with 7921 images and a test data set with 2007 images. Each image consists of handwritten digit comprising 16×16 pixels. Then the input data vector $\mathbf{x}_k$ has dimension 256 and is formed by row concatenation of the original image. The study also considers the influence of noise on the feature extraction process as well as the performance of the classifier. A Gaussian noise with variance of $\sigma^2 = 0.25$ will be added to each digit of the training and test sets. Figure 1 illustrates examples of digits and their noisy versions.

Fig. 1. Digits without and with noise

Basis vectors in feature space. The kernel function used to compute the dot products in feature space is given by a radial basis function (RBF)

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}) \tag{13}$$

Using this kernel function, a very efficient implementation for the incomplete Cholesky decomposition algorithm exists (accessible in [1]) having as input:

- the training data set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_K]$,
- the parameter of RBF kernel, σ
- a threshold to control approximation error of the decomposition.

The parameter σ controls the decay of the approximation error. The matrix $\mathbf{C}$ is formed iteratively, starting with one row up to R when the error is less than threshold. The error ϵ is approximated as $\epsilon \approx tr(\mathbf{K}_s - \mathbf{K}_{rs}^T \mathbf{K}_r^{-1} \mathbf{K}_{rs})$. It has to be noticed that using an RBF function, the trace is obtained as $tr(\mathbf{K}) = K$ where K denotes the size of the data set. The outputs of the algorithm are the index of the pivoting scheme and the matrix $\mathbf{C}$. The former allow to identify the subset $\mathbf{\Phi_R}$ which will contribute to form R orthogonal basis vectors (see eqn.12). In the experimental results to be discussed in the following, a threshold parameter was set to $\epsilon \leqslant 0.01K$ and the width parameter σ of the RBF kernel was set to different values. Furthermore the size of the training data set was varied to comprise 10%, 50% and 100% of the available data, respectively. Table 1 presents the size (R) of subset Φ_R for different sizes of the training set and the different values σ used in the simulations.

Notice that to compute the basis vectors (see eqn. 12), an eigendecomposition of matrix $\mathbf{Q}$ needs to be performed. The values of R obtained make it possible to achieve

[1] http://www.di.ens.fr/ fbach/kernel-ica/index.htm

Table 1. Size (R) of subset $\mathbf{\Phi}_R$ for different values of σ using training sets with different sizes(K)

K	100%				50%				10%			
σ	5	8	10	12	5	8	10	12	5	8	10	12
raw data	1807	241	91	-	1169	206	80	37	335	132	63	32
noisy data	-	1062	306	160	-	775	282	153	-	318	190	115

feasible eigendecompositions even when the size of the training set is prohibitively large, for example using with 50% or 100% of the available data. Besides that the size R influences other aspects of the application of the method to compute the kernel features like

- storage requirements to store the data training set that belongs to Φ_R
- dimension of the data in feature space is limited to R.

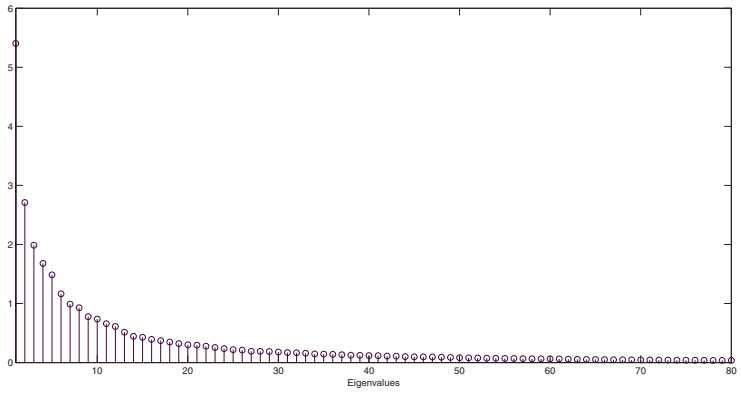

Fig. 2. Eigenvalues of the covariance matrix of the training set(without noise)

Basis vectors in input space. In the input space the basis vectors are computed using principal component analysis(PCA). The covariance matrix of the training data set is computed, and the basis vectors correspond to its eigenvectors. Ordering the eigenvectors of the covariance matrix according to their related eigenvalues, the basis vectors then represent the directions of maximal variance of the training data set. Fig.2 shows the eigenvalues of the covariance matrix of the complete training data set. It can be seen that the data mostly spread in 50 of the 256 directions of the input space.

Classification: nearest neighbor and linear discriminant. The USPS data set is a benchmark used in many works and the best results report an error rate in the range $0.04 - 0.05$. The complete training set was used to classify the test data set with a nearest-neighbor (NN) strategy using one nearest neighbor only, and an error rate equal to 0.056 was achieved. The same training set was also used to compute linear discriminant functions (RL) using a mean squared error criterion ([5],pp 239-268). Each digit

of the test set is then assigned to the class whose discriminant function has the larger value. For this linear classifier (RL), the error rate of the test data set is equal to 0.131.

Several simulations were conducted to evaluate the suitability of the projected data for classification. The data is projected onto the $L < R$ most significant directions that form the basis vectors and the values of the projections are used as input to the classifiers.

Figure 3 illustrates the performance of the k-NN classifier varying the number of projections. The classifiers trained with the complete training data set have the best performance, achieving an error rate of 0.05 while with the smaller data set the error rate is around 0.1. With PCA the best performance is achieved using 50 projections roughly. This result has to expected as the covariance matrix exhibits approximately 50 significant eigenvalues, the remaining eigenvalues are very close to zero. The best performance of k-NN classifier having PCA or kernel features have similar error rate 0.05. However, using $L > 50$ PCA projections the error rate increases slightly (0.007) while with the kernel features computed using $\sigma = 5$ the error rate is maintained (see table 2).

Figure 4 shows the results for the linear classifier (RL), and the performance here is less dependent on the size of the training set. The error rate of the classifier when the inputs are the projections in feature space is around 0.09 while with PCA projections it is 0.14. The optimal number of PCA projections is around 50, while with kernel methods more than 100 are needed. The improvement of the linear classifier with kernel projections is to be expected as in feature space the data should be linearly separable. Note that the k-NN classifier having as input the kernel projections and trained with 10% of the training data set shows a similar performance. The results presented in [9] show a similar tendency: the linear SVM classifier performs better using projections computed with KPCA instead of PCA. Calculating 2048 projections in KPCA feature space, the improvement in error rate amounts to 0.046 if a polynomial kernel is used. The data set used for training consisted of 3000 examples.

In kernel projective techniques, the number (R) of basis vectors has the highest value when $\sigma = 5$, but the performance of RL after $L = 100$, does not change increasing L.

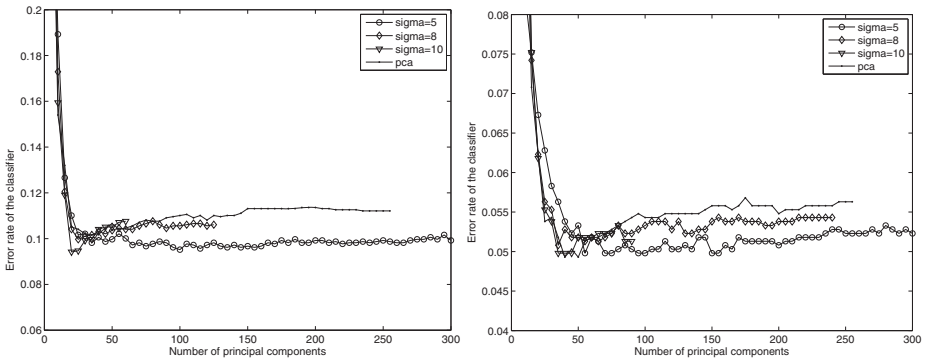

Fig. 3. Performance of NN using projections in input space (PCA) and in feature space. Training set with: 729(*left*) or 7291 (*right*) images.

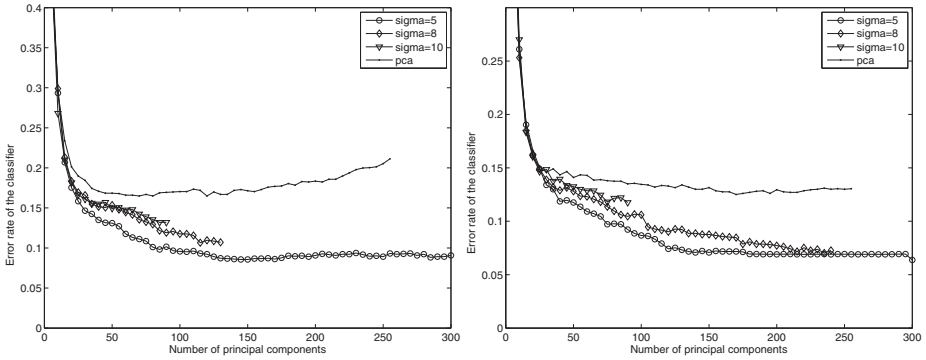

Fig. 4. Performance of the RL using projections in input space (PCA) and in feature space. Training set with: 729(*left*) or 7291 (*right*) images.

On the other hand, for the other values of σ the best performance is achieved using projections onto all the available basis vectors. All figures demonstrate that the RBF kernel with $\sigma = 5$ shows the best performance. But if noise is added, this parameter needs to be changed and best performance is obtained with $\sigma = 8$. Table 2 presents the classification error rate using a variable number of projections. The table also shows the performance of the system when Gaussian noise is added to both the training and test data sets. It is obvious that the performance of all classifiers degrades if noise is added to the data. The degradation level has similar values for both classifiers whatever is the number of projections used as input.

Table 2. Error rate of the classifiers using data (training and test) sets with and without noise

	K	100%				50%				10%			
	L	10	20	50	100	10	20	50	100	10	20	50	100
raw data	PCA - kNN	0.104	0.062	0.049	0.056	0.123	0.068	0.061	0.061	0.125	0.105	0.102	0.110
	PCA - RL	0.252	0.166	0.143	0.135	0.214	0.127	0.146	0.143	0.295	0.201	0.171	0.171
	RBF5 - kNN	0.083	0.063	0.054	0.050	0.099	0.069	0.061	0.061	0.132	0.109	0.101	0.097
	RBF5 - RL	0.193	0.157	0.120	0.086	0.184	0.111	0.125	0.087	0.235	0.176	0.134	0.095
noisy data	PCA - kNN	0.192	0.112	0.093	0.102	0.208	0.125	0.120	0.114	0.240	0.191	0.180	0.183
	PCA - RL	0.280	0.189	0.168	0.166	0.255	0.209	0.193	0.183	0.322	0.238	0.224	0.212
	RBF8 -kNN	0.212	0.121	0.097	0.105	0.195	0.133	0.120	0.106	0.191	0.172	0.153	0.170
	RBF8 - RL	0.438	0.195	0.165	0.162	0.238	0.197	0.172	0.169	0.290	0.221	0.215	0.204

5 Conclusion

In this paper we formulate the Nyström approach to low-rank approximations of the kernel matrix to be used as feature extraction. As the basis vector is expressed in terms of the training data set the method has the advantage of selecting a subset of the training set reducing that way the complexity of the problem during testing. However this

reduction depends on the parameter of RBF which must be carefully assigned in order to have a good performance without increasing the complexity. The feature extraction method was applied to the USPS data set in order to do classification. In what concerns classification, using the projections in feature space and a simple linear classifier the performance was good even when the training set have a reduced size. The improvement in performance has the same value as the one presented in [9]. However the method presented to compute kernel projections is less complex than the KPCA used in the referred work. Nevertheless this method needs to be applied to other data sets and having other linear classifiers in order to corroborate the conclusions drawn.

Acknowledgment

A.R. Teixeira received a PhD Scholarship (SFRH/BD/28404/2006) supported by the Portuguese Foundation for Science and Technology (FCT).

References

1. Achlioptas, D., McSherry, F., Schölkopf, B.: Sampling techniques for kernel methods. In: Advances in Neural Information Processing Systems, pp. 335–342. MIT Press, Cambridge (2002)
2. Bach, F.R., Jordan, M.I.: Kernel independent component analysis. Journal of Machine Learning Research 3, 1–48 (2002)
3. Baudat, G., Anouar, F.: Feature vector selection and projection using kernels. Neurocomputing 55, 21–38 (2003)
4. Cawley, G.C., Talbot, N.L.C.: Efficient formation of a basis in a kernel induced feature space. In: Verleysen, M. (ed.) European Symposium on Artificial Neural Networks, Bruges, Belgium, pp. 1–6 (2002)
5. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. (2001)
6. Fowlkes, C., Belongie, S., Chung, F., Malik, J.: Spectral grouping using the nyström method. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(2), 214–225 (2004)
7. Franc, V., Hlaváč, V.: Greedy algorithm for a training set reduction in the kernel methods. In: Petkov, N., Westenberg, M.A. (eds.) CAIP 2003. LNCS, vol. 2756, pp. 426–433. Springer, Heidelberg (2003)
8. Müller, K.-R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An introduction to kernel-based algorithms. IEEE Transactions on Neural Networks 12(2), 181–202 (2001)
9. Schölkopf, B., Smola, A., Müller, K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10, 1299–1319 (1998)
10. Williams, C.K.I., Seeger, M.: Using the nyström method to speed up kernel machines. In: Advances in Neural Information Processing Systems, pp. 682–688. MIT Press, Cambridge (2000)