Accelerating Kernel Neural Gas

Frank-Michael Schleif, Andrej Gisbrecht, and Barbara Hammer

CITEC centre of excellence, Bielefeld University, 33615 Bielefeld, Germany,
{fschleif|agisbrec|bhammer}@techfak.uni-bielefeld.de

Abstract. Clustering approaches constitute important methods for unsupervised data analysis. Traditionally, many clustering models focus on spherical or ellipsoidal clusters in Euclidean space. Kernel methods extend these approaches to more complex cluster forms, and they have been recently integrated into several clustering techniques. While leading to very flexible representations, kernel clustering has the drawback of high memory and time complexity due to its dependency on the full Gram matrix and its implicit representation of clusters in terms of feature vectors. In this contribution, we accelerate the kernelized Neural Gas algorithm by incorporating a Nyström approximation scheme and active learning, and we arrive at sparse solutions by integration of a sparsity constraint. We provide experimental results which show that these accelerations do not lead to a deterioration in accuracy while improving time and memory complexity.

1 Introduction

The dramatic growth in data generating applications and measurement techniques has created many high-volume data sets. Most of them are stored digitally and need to be efficiently analyzed to be of use. Clustering methods are very important in this setting and have been extensively studied in the last decades [9]. Challenges are mainly in time and memory efficient and accurate processing of such data with flexible and compact data analysis tools. The Neural Gas vector quantizer [13] (NG) constitutes a very effective prototype based clustering approach with a wide range of applications and extensions [21, 10, 1, 23]. It is well known for its initialization insensitivity, making it a valuable alternative to traditional approaches like k-means. It suffers, however, from its focus on spherical or ellipsoidal clusters such that complex cluster shapes can only be represented based on an approximation with a very large number of spherical clusters. Alternative strategies dealing with more complex data manifolds or novel metric adaptation techniques in clusterings are typically still limited, unable to employ the full potential of a complex modeling [7, 2]. The success of kernel methods in supervised learning tasks [20, 19] has motivated recent extensions of unsupervised schemes to kernel techniques, see e.g. [22, 3, 17, 5, 6].

Kernelized neural gas (KNG) was proposed in [17] as a non-linear, kernelized extention of the Neural Gas vector quantizer. While this approach is quite promising it has been used only rarely due to its calculation complexity which is roughly in $O(N^2)$, with N as the number of points. Drawbacks are given by the storage of a large kernel matrix and the update of a combinatorial coefficient

matrix, representing the prototypes implicitly. This makes the approach time and memory consuming already for small data sets.

Modern approaches in discriminative learning try to avoid the direct storage and usage of the full kernel matrix and restrict the underlying optimization problem to subsets thereof, see e.g. [16, 20]. For unsupervised kernel methods comparably few work has been done so far to overcome the memory and time complexity for large data sets [12]. For the KNG approach no such strategy has been proposed at all up to our knowledge.

In this contribution, we extend KNG towards a time and memory efficient method incorporating a variety of techniques: The Nyström-Approximation of Gram matrices constitutes a classical approximation scheme [25, 11], permitting the estimation of the kernel matrix by means of a low dimensional approximation. Further speedup can be achieved by using the explicit margin information to arrive at an active learning scheme. The high memory requirement which is caused by the implicit representation of prototypes in terms of feature vectors which, unlike for the supervised support vector machine, are usually not sparse, can be dealt with by incorporating sparsity constraints. Sparsity is a natural concept in the encoding of data [15] and can be used to obtain compact models. This concept has already been used in many machine learning methods [10, 8] and different measures of sparsity have been proposed [15, 8]. We integrate such a sparsity constraint into KNG.

In Section 2 we present a short introduction into kernels and give the notations used throughout the paper. Subsequently we present the KNG algorithm and the approximated variant, accelerated KNG (AKNG) by means of the Nyström approximation, active learning, and the additional sparsity constraint. We show the efficiency of the novel approach by experiments on several data sets. Finally, we conclude with a discussion in Section 4.

2 Preliminaries

We consider vectors $\mathbf{v}_j \in \mathbb{R}^d$, d denoting the dimensionality, n the number of samples. N prototypes $\mathbf{w}_i \in \mathbb{R}^d$ induce a clustering by means of their receptive fields which consist of the points $\mathbf{v}$ for which $d(\mathbf{v}, \mathbf{w_i}) \leq d(\mathbf{v}, \mathbf{w_j})$ holds for all $j \neq i$, d denoting a distance measure, typically the Euclidean distance.

A kernel function $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is implicitly induced by a feature mapping $\phi : \mathbb{R}^d \to \mathcal{F}$ into some possibly high dimensional feature space $\mathcal{F}$ such that

$$\kappa(\mathbf{v}_1, \mathbf{v}_2) = \langle \phi(\mathbf{v}_1), \phi(\mathbf{v}_2) \rangle_{\mathcal{F}} \tag{1}$$

holds for all vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, where the inner product in the feature space is considered. Hence κ is positive semi-definite. Using the linearity in the Hilbert-space, we can express dot products of elements of the linear span of ϕ of the form $\sum_i \alpha_i \phi(\mathbf{v}_i)$ and images $\phi(\mathbf{v})$ via the form $\sum_i \alpha_i \kappa(\mathbf{v}_i, \mathbf{v})$. This property is used in [17], to derive a kernelization of Neural Gas.

3 Neural Gas Algorithm

The Neural Gas (NG) algorithm is a type of vector quantizer providing a compact representation of the underlying data distributions [14]. Its goal is to find pro-

totype locations $\mathbf{w}_i$ such that these prototypes represent the data $\mathbf{v}$, distributed according to $\mathcal{P}$, as accurately as possible, minimizing the energy function:

$$E_{NG}(\gamma) = \frac{1}{C(\gamma, K)} \sum_{i=1}^{N} \int \mathcal{P}(\mathbf{v}) \cdot h_\gamma(\mathbf{v}_i, \mathbf{W}) \cdot (\mathbf{v} - \mathbf{w}_i)^2 \, d\mathbf{v} \tag{2}$$

with neighborhood function of Gaussian shape: $h_\gamma(\mathbf{v}_i, \mathbf{W}) = \exp(-k_i(\mathbf{v}, \mathbf{W})/\gamma)$. $k_i(\mathbf{v}, \mathbf{W})$ yields the number of prototypes $\mathbf{w}_j$ for which the relation $d(\mathbf{v}, \mathbf{w}_j) \leq d(\mathbf{v}, \mathbf{w}_i)$ is valid, i.e. the winner rank. $C(\gamma, K)$ is a normalization constant depending on the neighborhood range γ. The NG learning rule is derived thereof by stochastic gradient descent:

$$\triangle \mathbf{w}_i = \epsilon \cdot h_\gamma(\mathbf{v}_i, \mathbf{W}) \cdot (\mathbf{v} - \mathbf{w}_i) \tag{3}$$

with learning rate ϵ. Typically, the neighborhood range γ is decreased during training to ensure independence of initialization and optimization of the quantization error. NG is a simple and highly effective algorithm for data clustering.

3.1 Kernelized Neural Gas

We now briefly review the main concepts used in Kernelized Neural Gas (KNG) as given in [17]. KNG optimizes the same cost function as NG but with the Euclidean distance substituted by a distance induced by a kernel. Since the feature space is unknown, prototypes are expressed implicitly as linear combination of feature vectors $\mathbf{w}_i = \sum_{l=1}^{n} \alpha_{i,l} \phi(\mathbf{v}_l)$, $\alpha_i \in \mathbb{R}^n$ is the corresponding coefficient vector. Distance in feature space for $\phi(\mathbf{v}_j)$ and $\mathbf{w}_i$ is computed as:

$$d_{i,j}^2 = \|\phi(\mathbf{v}_j) - \mathbf{w}_i\|^2 = \|\phi(\mathbf{v}_j) - \sum_{l=1}^{n} \alpha_{i,l} \phi(\mathbf{v}_l)\|^2 \tag{4}$$

$$= k(\mathbf{v}_j, \mathbf{v}_j) - 2 \sum_{l=1}^{n} k(\mathbf{v}_j, \mathbf{v}_l) \cdot \alpha_{i,l} + \sum_{s,t=1}^{n} k(\mathbf{v}_s, \mathbf{v}_t) \cdot \alpha_{i,s} \alpha_{i,t} \tag{5}$$

The update rules of NG can be modified by substituting the Euclidean distance by the formula (4) and taking derivatives with respect to the coefficients $\alpha_{i,l}$. The detailed equations are available in [17].

3.2 Nyström Approximation of the Kernel Matrix

As pointed out in [25] different strategies have been proposed to overcome the complexity problem caused by the kernel matrix in modern machine learning algorithms. One promising approach is the Nyström approximation.

It originates from the numerical treatment of integral equations of the form $\int \mathcal{P}(y) k(x, y) \phi_i(y) dy = \lambda_i \phi_i(x)$ where $\mathcal{P}(\cdot)$ is the probability density function, k is a positive definite kernel function, and $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$ and $\phi_1, \phi_2, \ldots$ are the eigenvalues and eigenfunctions of the integral equation. Given a set of i.i.d. samples $\{x_1, \ldots, x_q\}$ drawn from $\mathcal{P}(\cdot)$, the basic idea is to approximate the

integral by the empirical average $1/q \sum_{j=1}^{q} k(x, x_j)\phi_i(x_j) \approx \lambda_i \phi_i(x)$ which can be written as the eigenvalue decomposition: $K\phi = q\lambda\phi$ where $K_{q\times q} = [K_{i,j}] = [k(x_i, x_j)]$ is the kernel matrix defined on X, and $\phi = [\phi_i(x_j)] \in \mathbb{R}^q$. Solving this equation we can calculate $\phi_i(x)$ as $\phi_i(x) \approx 1/(q\lambda) \sum_{j=1}^{q} k(x, x_j)\phi_i(x_j)$ which is costly. To reduce the complexity, one may use only a subset of the samples which is commonly known as the Nystöm method.

Suppose the sample set $V = \{\mathbf{v}_i\}_{i=1}^{n}$, with the corresponding $n \times n$ kernel matrix K. We randomly choose a subset $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^{q}$ of landmark points and a corresponding kernel sub matrix $\mathbf{Q}_{q\times q} = [k(\mathbf{z}_i, \mathbf{z}_j)]_{i,j}$. We calculate the eigenvalue decomposition of this sub matrix: $\mathbf{Q}\phi_z = q\lambda_z\phi_z$ and obtain the corresponding eigenvector $\phi_z \in \mathbb{R}^q$ and the eigenvalue $q\lambda_z$. Subsequently we calculate the interpolation matrix $\hat{\mathbf{K}}_{n\times q} = [k(\mathbf{v}_i, \mathbf{z}_j)]_{i,j}$ to extend the result to the whole set V. We approximate the eigen-system of the full $K\phi_K = \phi_K\lambda_K$ by [24]:

$$\phi_K \approx \sqrt{\frac{q}{n}}\hat{\mathbf{K}}\phi_Z\lambda_{\mathbf{Z}}^{-1}, \lambda_K \approx \frac{n}{q}\lambda_{\mathbf{Z}}$$

K can be subsequently reconstructed as

$$K \approx \left(\sqrt{\frac{q}{n}}\hat{\mathbf{K}}\phi_Z\lambda_{\mathbf{Z}}^{-1}\right)\left(\frac{n}{q}\lambda_{\mathbf{Z}}\right)\left(\sqrt{\frac{q}{n}}\hat{\mathbf{K}}\phi_Z\lambda_{\mathbf{Z}}^{-1}\right)' = \hat{\mathbf{K}}\mathbf{Q}^{-1}\hat{\mathbf{K}}'$$

To integrate the Nyström approximation into KNG we only need to modify the distance calculation between a prototype $\mathbf{w}_i$ and a data point $\phi(\mathbf{v}_j)$ accordingly. The original update equation for the coefficient matrix in KNG reads as:

$$\alpha_{j,l}^{t+1} = \begin{cases} [1 - \epsilon \cdot h_\gamma(k_j(\phi(\mathbf{v}_i), \mathbf{W}))] \cdot \alpha_{j,l}^{t} & \text{if } \mathbf{v}_i \neq \mathbf{v}_l \\ [1 - \epsilon \cdot h_\gamma(k_j(\phi(\mathbf{v}_i), \mathbf{W}))] \cdot \alpha_{j,l}^{t} + \epsilon \cdot h_\gamma(k_j(\phi(\mathbf{v}_i), \mathbf{W})) & \text{if } \mathbf{v}_i = \mathbf{v}_l \end{cases}$$

with $t+1$ indicating the time step and $\mathbf{w}_k \in \mathbf{W}$ defined as in Eq. 4. The distance calculation using the Nyström approximation is done as follows: (6):

$$d_{\cdot,j} = K(j,j) - 2 \cdot T_{\cdot,j} + \text{diag}(\psi \cdot T') \tag{6}$$

$$\text{with } T_{i,\cdot} = ((\alpha_i \cdot \hat{\mathbf{K}}) \cdot \mathbf{Q}^{-1}) \cdot \hat{\mathbf{K}}' \tag{7}$$

where diag provides the main diagonal elements of the associated matrix. With Nyström-approximation the complexity is reduced to $O(q^2 N)$ [24].

3.3 Sparse Coefficient Matrix

In [15] sparsity has been found to be a natural concept in the visual cortex of mammals. This work motivated the integration of sparsity concepts into many machine learning methods to obtain sparse but efficient models. Here we will integrate sparsity as an additional constraint on the coefficient matrix α such that the amount of non-zero coefficients is limited. This leads to a compact description of the prototypes by means of sparse linear mixture models. We use the sparsity measure given in [15]. The sparsity $\mathbb{S}$ of a row of α is measured as

$$\mathbb{S}(\alpha_i) = -\sum_{l} S\left(\frac{\alpha_{i,l}}{\sigma}\right) \tag{8}$$

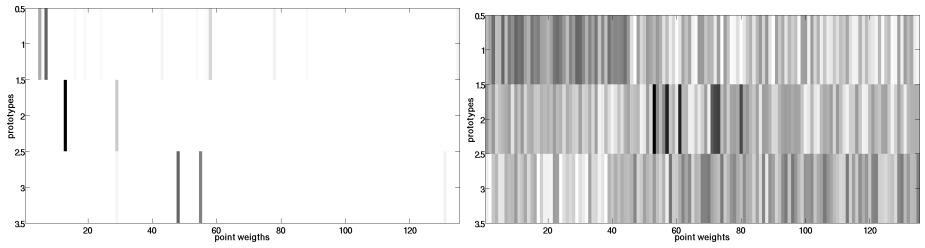

Fig. 1. Effect of the sparsity constraint for DS2 shown by means of the γ-matrix (normalized for better comparison) showing point weights (x-axis) for each prototype (y-axis). With sparsity left and without right. Dark values (1) indicated high loaded or high lighted data points for the considered prototype in the γ matrix. Data points with very low values (0) over all prototypes can be safely removed from the model.

with σ as a scaling constant. The function S can be of different type, here we use $S(x) = \log(1 + x^2)$. We change the energy function of the KNG as follows

$$E_{KNG}(\gamma) = \frac{1}{C(\gamma, K)} \sum_{i=1}^{N} \int \mathcal{P}(\phi(\mathbf{v}) \cdot h_\gamma(\phi(\mathbf{v}_i), \mathbf{W}) \cdot \|\phi(\mathbf{v}) - \mathbf{w}_i\|^2 \, d\phi(\mathbf{v}) \\ -\beta \cdot \mathbb{S}(\alpha_i)$$

The updates for the coefficients of $\mathbf{w}_i$ are exactly the same as for the standard KNG using the Nyström formula to approximate the Gram matrix and including the additional term caused by the sparsity constrained

$$\frac{\partial \mathbb{S}}{\partial \alpha_{i,l}} = -\frac{2/\sigma^2 \cdot \alpha_{i,l}}{1 + (\alpha_{i,l}/\sigma)^2}$$

In addition, we enforce the constrained $\alpha_{i,l} \in [0, 1]$ and $\sum_l \alpha_{i,l} = 1$ for better interpretability. The effect of the sparsity constraint on the UCI iris data is shown in Figure 3.3 with 1 prototype per class. The sparse model has an accuracy of $\approx 90\%$ whereas the original solution achieves only 86%.

3.4 Active Learning

The sparse coefficients α give rise to an active learning scheme in a similar manner as proposed in [18], leading to faster learning. The matrix α encodes a weighting of the data-points. We take the column-wise mean of α as $\bar{\alpha}$ and calculate a threshold δ for each data-point indicating its relative importance for the learning. The average weight for a data-point in α is given as $\delta^* = \alpha_{i,j} = 1/N$, due to the normalization constraint. Weights close to this point are not sufficiently learned, such that they have not been deleted or emphasized so far. We transfer these weights to a skip probability for each data-point using:

$$\delta = 1/2 \cdot \exp\left(-\frac{(\bar{\alpha_j} - \delta^*)^2}{(2 \cdot std(\bar{\alpha})^2)}\right) \quad \text{with std - as the standard deviation} \qquad (9)$$

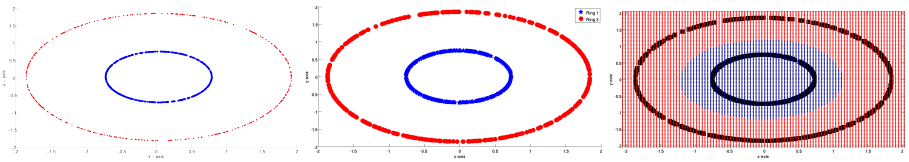

Fig. 2. Ring data set (left), post-labeled KGLVQ model (middle), the outer ring is red ('o'), the inner ring is blue ($\star$). The plot on the right shows the cluster boundaries of the model from the middle plot. The model was calculated without sparsity. It can be clearly seen that the A-KNG with an rbf kernel sucessfully separated these two clusters, with good cluster boundaries and a large margin between the two rings.

This denotes the probability of the data-point to be skipped during learning. It should be noted, that at the beginning of the learning α is initialized randomly such that the probability of a data-point to be skipped is random; during learning only those points are likely to be skipped which are either not or most relevant for the model. In this line we roughly learn the model by considering an ϵ-tube around the cluster centers. However, by taking the probability concept *all* points are taken into account albeit with probably small probability.

4 Experiments

As a proof of concept, we start with a toy data set (DS1) and an RBF kernel. The data consist of 800 data points with 400 per ring in 2 dimensions (x/y) as shown in Figure 2. The first ring has a radius of $r = 10$ and the second $r = 4$, points are randomly sampled in $[0, 2\pi]$. The data set has been normalized in $N(0, 1)$. We also analyzed the ring data using the additional sparsity constraint. In the original model 53% of the weights, averaged over the prototypes are almost 0 (values $\leq 1e - 5$). In the sparsity approach we used σ^2 as the variance of the data scaled by 0.01 and $\beta = 1$ and obtained a model with about 75% of the points close to zero.

Following the work given in [12] we analyze several UCI benchmarks. We consider the well known iris data set (DS2), the Wisconsin Breast cancer data (WBC) (DS3), the Spam database (DS4), and Pima diabetes (DS5). Details about the data can be found in [4]. DS2 consists of 150 instances in three groups and is known to be almost linear separable by two hyperplanes. DS3 consists of 683 items. For this dataset non-linear supervised learning methods have been found to be very effective whereas linear approaches are not so effective. This motivates the assumption that kernelization might prove beneficial. The data set DS4 contains 1534 samples, and classification is difficult. The fifth data set (DS5) contains 768 samples. For each data set we used one prototype per expected group. The results are shown in Table 4.

All results are obtained using 10 cycles with a Nyström approximation of $1-10\%$ of the original kernel matrix, $\beta \in [0.001, 10]$, and the sparsity $\sigma \in [1, 100]$ determined on an independent test set. The value of the Nyström approximation is not very critical for the accuracy and mainly influences the runtime performance, whereas a too sparse solution can lead to a decrease in accuracy. Dataset

Table 1. Post labeled accuracy vs. runtime over 10 runs. For AK-NG and K-NG 10 cycles are calculated, each. Best results in bold, *-ed results are taken from [12]

Algorithm	Iris data	WBC	Spam	Pima diabetes
NG	91.7%/n.a.*	96.1%/n.a.*	68.4%/n.a. *	70.31%/7s
K-NG	90.0%/2.6s	91.7%/5.77s	**86.5**%/350s	71.74%/21s
AK-NG	92.6%/**0.14s**	92.1%/**0.73s**	84.42%/**2.9s**	**73.05**%/**0.94s**
K-Grower	**94.7**%/12.95s^*	**97.0**%/807.17s^*	81.3%/ $\gg 1000s^*$	n.a.
SUK-Grower	93.4%/47.95s^*	96.8%/22.85s^*	80.2%/44.83s^*	n.a.

D2,D3, and D5 are analyzed using an RBF kernel with a $\sigma^2 = \{1, 0.01, 0.1\}$ respectively, for DS4 we used a linear kernel. The other experimental settings have been chosen in accordance to [12] for compatibility. We also report two alternative state of the art clustering methods by means of core sets provided in [12], referred to as K-Grower and SUK-Grower. Analyzing the results given in Table 4 the AK-NG is significantly faster in calculating the clustering models than all other approaches with the same or only slightly less accuracy. Analyzing the optimizations separately for $DS3 - DS5$ we find: sparsity leads to a reduced memory consumption of $\approx 25\%(DS3), \approx 30\%(DS4)$ and $\approx 41\%(DS5)$ with respect to the unoptimized approach; Nyström approximation leads to a speedup of $\approx$ 1.6($DS3$), $\approx$ 6.8($DS4$) and $\approx$ 2($DS5$) and the active learning strategy behaves similar. The effect of these optimizations has almost no effect on the accuracy, giving appropriate parameters as pointed out before.

5 Conclusions

In this paper we proposed an extension of kernelized neural gas with a significantly reduced model complexity and time complexity by incorporating the Nyström approximation and a sparsity constraint. We compared the efficiency of our approach with alternative state of the art clusterings with respect to clustering accuracy as well as efficiency. We found the AK-NG is similarly effective and significantly faster with respect to the considered approaches. So far, we tested the algorithm on UCI benchmarks, its application to real life very large data sets being the subject of ongoing work.

Acknowledgment This work has been supported by the German Res. Found. (DFG), HA2719/4-1 (Relevance Learning for Temporal Neural Maps) and in the frame of the centre of excellence 'Cognitive Interaction Technologies'.

References

1. Ardizzone, E., Chella, A., Rizzo, R.: Color image segmentation based on a neural gas network. In: Marinaro, M., Morasso, P.G. (eds.) Proc. ICANN'94, Int. Conf. on Artif. Neural Netw. vol. II, pp. 1161–1164. Springer, London, UK (1994)
2. Arnonkijpanich, B., Hasenfuss, A., Hammer, B.: Local matrix adaptation in topographic neural maps. Neurocomputing 74(4), 522–539 (2011)

3. Ben-Hur, A., Horn, D., Siegelmann, H., Vapnik, V.: Support vector clustering. Journal of Machine Learning Research 2, 125–137 (2001)
4. Blake, C., Merz, C.: UCI repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science, available at: http://www.ics.uci.edu/ mlearn/MLRepository.html (1998)
5. Filippone, M., Camastra, F., Massulli, F., Rovetta, S.: A survey of kernel and spectral methods for clustering. Pattern Recognition 41, 176–190 (2008)
6. Hammer, B., Hasenfuss, A.: Topographic mapping of large dissimilarity datasets. Neural Computation 22(9), 2229–2284 (2010)
7. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer, New York (2001)
8. Hoyer, P.: Non-negative Matrix Factorization with sparseness constraints. Journal of Machine Learning Research 5, 1457–1469 (2004)
9. Jain, A.K.: Data clustering: 50 years beyond K-means. PatRecL 31, 651–666 (2010)
10. Labusch, K., Barth, E., Martinetz, T.: Learning data representations with sparse coding neural gas. In: Verleysen, M. (ed.) Proceedings of the European Symposium on Artificial Neural Networks ESANN. pp. 233–238. d-side publications (2008)
11. Li, M., Kwok, J., Lu, B.L.: Making large-scale nyström approximation possible. In: Proc. of the Int. Conf. on Mach. Learn. (ICML)'2010 (2009)
12. Liang, C., Xiao-Ming, D., Sui-Wu, Z., Yong-Qing, W.: Scaling up kernel grower clustering method for large data sets via core-sets. Acta Automatica Sinica 34(3), 376–382 (2008)
13. Martinetz, T., Schulten, K.: A "Neural-Gas" network learns topologies. In: Kohonen, T., Mäkisara, K., Simula, O., Kangas, J. (eds.) Proc. International Conference on Artificial Neural Networks (Espoo, Finland). vol. I, pp. 397–402. North-Holland, Amsterdam, Netherlands (1991)
14. Martinetz, T.M., Berkovich, S.G., Schulten, K.J.: 'Neural-gas' network for vector quantization and its application to time-series prediction. IEEE Trans. on Neural Networks 4(4), 558–569 (1993)
15. Olshausen, B., Field, D.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Letters to Nature 381, 607–609 (1996)
16. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. pp. 185–208. MIT Press, Cambridge, MA, USA (1999)
17. Qin, A.K., Suganthan, P.N.: A novel kernel prototype-based learning algorithm. In: Proc. of ICPR'04. pp. 2621–624 (2004)
18. Schleif, F.M., Hammer, B., Villmann, T.: Margin based active learning for lvq networks. Neurocomputing 70(7-9), 1215–1224 (2007)
19. Schlkopf, B., Smola, A.: Learning with Kernels. MIT Press (2002)
20. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis and Discovery. Cambridge University Press (2004)
21. Strickert, M., Hammer, B.: Neural gas for sequences. In: Proc. International Workshop on Self-Organizing Maps (WSOM'2003). pp. 53–58. Kitakyushu (2003)
22. Tsang, I., Kwok, J.: Distance metric learning with kernels. In: Kaynak, O. (ed.) Proc. of ICANN'2003. pp. 126–129. Istanbul (2003)
23. Walter, J.A., Martinetz, T.M., Schulten, K.J.: Industrial robot learns visuo-motor coordination by means of 'neural gas' network. In: Kohonen, T., Mäkisara, K., Simula, O., Kangas, J. (eds.) Artificial Neural Networks. vol. I, pp. 357–364. North-Holland, Amsterdam, Netherlands (1991)
24. Williams, C., Seeger, M.: Using the nystroem method to speed up kernel machines. In: Advances in Neural Information Processing Systems 13, pp. 682–688 (2001)
25. Zhang, K., Tsang, I., Kwok, J.: Improved nyström low-rank approximation and error analysis o. In: Proc. of the Int. Conf. on Mach. Learn. (ICML)'2010 (2009)