International Journal of Neural Systems, Vol. 0, No. 0 (April, 2000) 00–00

Linear time relational prototype based learning

Andrej Gisbrecht
Dept. of Techn., Univ. of Bielefeld, Universitätsstrasse 21-23
33615 Bielefeld, Germany
E-mail: agisbrec@techfak.uni-bielefeld.de

Bassam Mokbel
Dept. of Techn., Univ. of Bielefeld, Universitätsstrasse 21-23
33615 Bielefeld, Germany
E-mail: bmokbel@techfak.uni-bielefeld.de

Frank-Michael Schleif
Dept. of Techn., Univ. of Bielefeld, Universitätsstrasse 21-23
33615 Bielefeld, Germany
E-mail: fschleif@techfak.uni-bielefeld.de

Xibin Zhu
Dept. of Techn., Univ. of Bielefeld, Universitätsstrasse 21-23
33615 Bielefeld, Germany
E-mail: xzhu@techfak.uni-bielefeld.de

Barbara Hammer*
Dept. of Techn., Univ. of Bielefeld, Universitätsstrasse 21-23
33615 Bielefeld, Germany
E-mail: bhammer@techfak.uni-bielefeld.de

Received (to be inserted
Revised by Publisher)

Prototype based learning offers an intuitive interface to inspect large quantities of electronic data in supervised or unsupervised settings. Recently, many techniques have been extended to data described by general dissimilarities rather than Euclidean vectors, so-called relational data settings. Unlike the Euclidean counterparts, the techniques have quadratic time complexity due to the underlying quadratic dissimilarity matrix. Thus, they are infeasible already for medium sized data sets. The contribution of this article is twofold: on the one hand we propose a novel supervised prototype based classification technique for dissimilarity data based on popular learning vector quantization, on the other hand we transfer a linear time approximation technique, the Nyström approximation, to this algorithm and an unsupervised counterpart, the relational generative topographic mapping. This way, linear time and space methods result. We evaluate the techniques on three examples from the biomedical domain.

1. INTRODUCTION

In many application areas such as bioinformatics, technical systems, or the web, electronic data sets are increasing rapidly with respect to size and complexity. Machine learning has revolutionized the possibility to deal with large electronic data sets in these areas by offering powerful tools to automatically extract a regularity from given data. Popular approaches provide diverse techniques for data struc-

*corresponding author

turing and data inspection. Visualization, clustering, or classification still constitute one of the most common tasks in this context [3,12,37,30].

Topographic mapping such as offered by the self-organizing map (SOM) [18] and its statistic counterpart, the generative topographic mapping (GTM) [6] provide simultaneous clustering and data visualization. For this reason, topographic mapping constitutes a popular tool in diverse areas ranging from remote sensing or biomedical domains up to robotics or telecommunication [18,21]. As an alternative, learning vector quantization (LVQ) represents priorly given classes in terms of labeled prototypes [18]. Learning typically takes place by means of Hebbian and anti-Hebbian updates. Original LVQ is based on heuristic grounds while modern alternatives are typically derived from an underlying cost function [33]. Similar to its unsupervised counterpart, LVQ has been successfully applied in diverse areas including telecommunication, robotics, or biomedical data analysis [18,2].

Like many classical machine learning techniques, GTM and LVQ have been proposed for Euclidean vectorial data. Modern data are often associated to dedicated structures which make a representation in terms of Euclidean vectors difficult: biological sequence data, text files, XML data, trees, graphs, or time series, for example [31,34]. These data are inherently compositional and a feature representation leads to information loss. As an alternative, a dedicated dissimilarity measure such as pairwise alignment, or kernels for structures can be used as the interface to the data In such cases, machine learning techniques which can deal with pairwise similarities or dissimilarities have to be used [24].

Also kernel methods like the Support Vector Machine (SVM) (see e.g.[9]) can be used for dissimilarity data, but complex preprocessing steps are necessary as discussed in the following. Kernel methods are known to be very effective, with respect to the generalization ability and, using modern approximation schemes, are also reasonable effective for larger data sets. In contrast to prototype methods the cost function is formulated typically by means of a convex problem, such that standard and effective optimization techniques can be used. Often they automatically adapt the model complexity, e.g. by means of support vectors for SVM, in accordance to the given *supervised* problem, which is often not the case for prototype methods. This strong framework however, requires a valid positive semi-definite kernel as an input, which is often not directly available for dissimilarity data. In fact, as discussed in the work of Pekalska[25], dissimilarity data can encode information in the euclidean and non-euclidean space and transformations to obtain a valid kernel may be inappropriate[32].

Quite a few extensions of prototype-based learning towards pairwise similarities or dissimilarities have been proposed in the literature. Some are based on a kernelization of existing approaches [7,39,29], while others restrict the setting to exemplar based techniques [10,19]. Some techniques build on alternative cost functions and advanced optimization methods [35,15]. A very intuitive method which directly extends prototype based clustering to dissimilarity data has been proposed in the context of fuzzy clustering [17] and later been extended to topographic mapping such as SOM and GTM [16,14]. Due to its direct correspondence to standard topographic mapping in the Euclidean case, we will focus on the latter approach. We will exemplarily look at this relational extension of GTM to investigate the performance of unsupervised prototype-based techniques for dissimilarity data. In this contribution, we will propose, as an alternative, a novel supervised prototype based classification scheme for dissimilarity data, with initial work given in [28]. Essentially, a modern LVQ formulation which is based on a cost function will be extended using the same trick to assess relational data.

One drawback of machine learning techniques for dissimilarities is given by their high computational costs: since they depend on the full (quadratic) dissimilarity matrix, they have squared time complexity; further, they require the availability of the full dissimilarity matrix, which is even the more severe bottleneck if complex dissimilarities such as e.g. alignment techniques are used. This fact makes the methods unsuitable already for medium sized data sets.

Here, we propose a popular approximation technique to speed up prototype based methods for dissimilarities: the Nyström approximation has been proposed in the context of kernel methods as a low rank approximation of the matrix [38]. In [13], preliminary work extends these results to dissimilarities. In this contribution, we demonstrate that the technique provides a suitable linear time approximation for GTM and LVQ for dissimilarities.

Now we first shortly recall the classical GTM and a variant of LVQ. Then we introduce the general concept underlying relational data representation, and we transfer this principle to GTM (shortly summarizing the results already presented in [14]) and to LVQ. The latter gives the novel algorithm relational generalized learning vector quantization. We recall the derivation of the low rank Nyström approximation for similarities and transfer this principle to dissimilarities. Linear time techniques for relational GTM and relational LVQ result. We demonstrate the behavior of the techniques in applications from the biomedical domain.

2. TOPOGRAPHIC MAPPING

Generative Topographic Mapping (GTM) has been proposed in [6] as a probabilistic counterpart to SOM. It models given data $\mathbf{x}^i \in \mathbb{R}^n$ by a constraint mixture of Gaussians induced by a low dimensional latent space. More precisely, regular lattice points $\mathbf{w}$ are fixed in latent space and mapped to target vectors $\mathbf{w} \mapsto \mathbf{t} = y(\mathbf{w}, \mathbf{W})$ in the data space, where the function y is typically chosen as generalized linear regression model $y : \mathbf{w} \mapsto \Phi(\mathbf{w}) \cdot \mathbf{W}$. The base functions Φ could be chosen as any set of nonlinear functions. Typically, equally spaced Gaussians with bandwidth σ are taken.

These prototypes in data space give rise to a constraint mixture of Gaussians in the following way. Every latent point induces a Gaussian

$$p(\mathbf{x}|\mathbf{w}, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{\frac{n}{2}} \exp\left(-\frac{\beta}{2}\|\mathbf{x} - y(\mathbf{w}, \mathbf{W})\|^2\right) \tag{1}$$

A mixture of K modes $p(\mathbf{x}|\mathbf{W}, \beta) = \sum_{k=1}^{K} \frac{1}{K} p(\mathbf{x}|\mathbf{w}^k, \mathbf{W}, \beta)$ is generated. GTM training optimizes the data log-likelihood with respect to $\mathbf{W}$ and β. This can be done by an EM approach, iteratively computing responsibilities

$$R_{ki}(\mathbf{W}, \beta) = p(\mathbf{w}^k|\mathbf{x}^i, \mathbf{W}, \beta) = \frac{p(\mathbf{x}^i|\mathbf{w}^k, \mathbf{W}, \beta)}{\sum_{k'} p(\mathbf{x}^i|\mathbf{w}^{k'}, \mathbf{W}, \beta)} \tag{2}$$

of component k for point $\mathbf{x}^i$, and optimizing model parameters by means of the formulas

$$\mathbf{\Phi}^T \mathbf{G}_{\text{old}} \mathbf{\Phi} \mathbf{W}_{\text{new}}^T = \mathbf{\Phi}^T \mathbf{R}_{\text{old}} \mathbf{X} \tag{3}$$

for $\mathbf{W}$, where $\mathbf{\Phi}$ refers to the matrix of base functions Φ evaluated at the lattice points $\mathbf{w}^k$, $\mathbf{X}$ refers to the data points, $\mathbf{R}$ to the responsibilities, and $\mathbf{G}$ is a diagonal matrix with accumulated responsibilities $G_{ii} = \sum_i R_{ki}(\mathbf{W}, \beta)$. The bandwidth is given by

$$\frac{1}{\beta_{\text{new}}} = \frac{1}{ND} \sum_{k,i} R_{ki}(\mathbf{W}_{\text{old}}, \beta_{\text{old}}) \|\Phi(\mathbf{w}^k)\mathbf{W}_{\text{new}} - \mathbf{x}^i\|^2 \tag{4}$$

where D is the data dimensionality and N the number of points. GTM is initialized by aligning the lattice image and the first two data principal components.

3. LEARNING VECTOR QUANTIZATION

As before, data $\mathbf{x}^i \in \mathbb{R}^n$ are given. Here we consider the crisp setting. That means, prototypes $\mathbf{w}^j \in \mathbb{R}^n, j = 1, \ldots, K$ in the data space decompose data into receptive fields $R(\mathbf{w}^j) := \{\mathbf{x}^i : \forall k\ d(\mathbf{x}^i, \mathbf{w}^j) \leq d(\mathbf{x}^i, \mathbf{w}^k)\}$ based on the squared Euclidean distance $d(\mathbf{x}^i, \mathbf{w}^j) = \|\mathbf{x}^i - \mathbf{w}^j\|^2$.

For supervised learning, data $\mathbf{x}^i$ are equipped with class labels $c(\mathbf{x}^i) \in \{1, \ldots, L\} = \mathcal{L}$. Similarly, every prototype is equipped with a priorly fixed label $c(\mathbf{w}^j)$. Let $\mathbb{W}_c = \{\mathbf{w}^\mathbf{l} | c(\mathbf{w}^l) = c\}$ be the subset of prototypes assigned to class $c \in \mathcal{L}$. A data point is classified according to the class of its closest prototype. The classification error of this mapping is given by the term $\sum_j \sum_{\mathbf{x}^i \in R(\mathbf{w}^j)} \delta(c(\mathbf{x}^i) \neq c(\mathbf{w}^j))$ with the delta function δ. This cost function cannot easily be optimized explicitly due to vanishing gradients and discontinuities. Therefore, LVQ relies on a reasonable heuristic by performing Hebbian updates of the prototypes, given a data point [18]. Recent alternatives derive similar update rules from explicit cost functions which are related to the classification error, but display better numerical properties such that efficient optimization algorithms can be derived thereof [33,26,36].

We introduce two special notations for the prototype which is closest to a given point $\mathbf{x}^i$ with the same label: $\mathbf{w}^+$ or a different label: $\mathbf{w}^-$. The corresponding distance d_i^+, d_i^-:

$$\begin{array}{rcl} d_i^+ & = & d(\mathbf{w}^+, \mathbf{x}^i) \text{ with } \mathbf{w}^+ \in \mathbb{W}_c,\ c = c(\mathbf{x}^i), \\ \mathbf{w}^+ & := & \mathbf{w}^l : d(x^i, w^l) \leq d(x^i, w^j), \{\mathbf{w}^j, \mathbf{w}^l\} \in \mathbb{W}_c \\ d_i^- & = & d(\mathbf{w}^-, \mathbf{x}^i) \text{ with } \mathbf{w}^- \notin \mathbb{W}_c,\ c = c(\mathbf{x}^i) \\ \mathbf{w}^- & := & \mathbf{w}^l : d(x^i, w^l) \leq d(x^i, w^j), \{\mathbf{w}^j, \mathbf{w}^l\} \notin \mathbb{W}_c \end{array}$$

Generalized LVQ [26] is derived from a cost function which can be related to the generalization ability of LVQ classifiers [33]:

$$E_{\text{GLVQ}} = \sum_i f\left(\frac{d_i^+ - d_i^-}{d_i^+ + d_i^-}\right) \tag{5}$$

where f is a differentiable monotonic function such as the hyperbolic tangent. Hence, for every data point, its contribution to the cost function is small if and only if the distance to the closest prototype with a correct label is smaller than the distance to a wrongly labeled prototype, resulting in a correct classification of the point and, at the same time, aiming at a large hypothesis margin of the classifier, i.e., a good generalization ability.

A learning algorithm can be derived thereof by means of standard gradient techniques. After presenting data point $\mathbf{x}^i$, its closest correct and wrong prototype, respectively, are adapted according to the prescription:

$$\begin{array}{rcl} \Delta\mathbf{w}^+(\mathbf{x}^i) & \sim & -f'(\mu(\mathbf{x}^i)) \cdot \mu^+(\mathbf{x}^i) \cdot \nabla_{\mathbf{w}^+(\mathbf{x}^i)} d_i^+ \\ \Delta\mathbf{w}^-(\mathbf{x}^i) & \sim & f'(\mu(\mathbf{x}^i)) \cdot \mu^-(\mathbf{x}^i) \cdot \nabla_{\mathbf{w}^-(\mathbf{x}^i)} d_i^- \end{array}$$

where

$$\mu(\mathbf{x}^i) = \frac{d_i^+ - d_i^-}{d_i^+ + d_i^-},$$

$$\mu^+(\mathbf{x}^i) = \frac{2 \cdot d_i^-}{(d_i^+ + d_i^-)^2},$$

$$\mu^-(\mathbf{x}^i) = \frac{2 \cdot d_i^+}{(d_i^+ + d_i^-)^2}.$$

For the squared Euclidean norm, the derivative yields

$$\nabla_{\mathbf{w}^j} d(\mathbf{x}^i, \mathbf{w}^j) = -2(\mathbf{x}^i - \mathbf{w}^j),$$

leading to Hebbian update rules of the prototypes which take into account the priorly known class information.

GLVQ constitutes one particularly efficient method to adapt the prototypes according to a given labeled data sets. Alternatives can be derived based on a labeled Gaussian mixture model, see e.g. [36]. Since the latter can be highly sensitive to model meta-parameters [5], we focus on GLVQ.

4. DISSIMILARITY DATA

Due to improved sensor technology or dedicated data formats, for example, data are becoming more and more complex in many application domains. To account for this fact, data are often addressed by a dedicated dissimilarity measure which respects the structural form of the data such as alignment techniques for bioinformatics sequences, functional norms for mass spectra, or the compression distance for texts [8]. The work in [25] is focused on the theoretical analysis of dissimilarity data and pseudo-euclidean data spaces and motivated our proposed method.

Prototype-based techniques such as GLVQ are restricted to Euclidean vector spaces such that their suitability for complex non-Euclidean data sets is highly limited. Here we propose an extension of GLVQ to general dissimilarity data.

We assume that data $\mathbf{x}^i, i = 1, \ldots, N$ are characterized by pairwise dissimilarities $d_{ij} = d(\mathbf{x}^i, \mathbf{x}^j)$. N denotes the number of data points. D refers to the corresponding dissimilarity matrix in $\mathbb{R}^{N \times N}$. We assume symmetry $d_{ij} = d_{ji}$ and zero diagonal $d_{ii} = 0$. However, D need not correspond to Euclidean data vectors, i.e. it is not guaranteed that data vectors $\mathbf{x}^i$ can be found with $d_{ij} = \|\mathbf{x}^i - \mathbf{x}^j\|^2$.

For every dissimilarity matrix D of this form, an associated similarity matrix is induced by $S = -JDJ/2$ where $J = (I - \mathbf{1}\mathbf{1}^T/N)$ with identity matrix I and vector of ones $\mathbf{1}$. D is Euclidean if and only if S is positive semi-definite (pdf). In general, S displays eigenvectors with p positive eigenvalues,

q negative eigenvalues, and $N-p-q$ eigenvalues 0, $(p,q,N-p-q)$ is referred to as the signature.

For kernel methods such as SVM, a correction of the matrix S is necessary to guarantee pdf. Three different techniques are very popular: the spectrum of the matrix S is changed, possible operations being clip (negative eigenvalues are set to 0), flip (absolute values are taken), or shift (a summand is added to all eigenvalues) [8]. Interestingly, some operations such as shift do not affect the location of local optima of important cost functions such as the quantization error [20], albeit the transformation can severely affect the performance of optimization algorithms [16]. As an alternative, data points can be treated as vectors which coefficients are given by the pairwise similarity. These vectors can be processed using standard, e.g. linear or Gaussian kernels. In [8] an extensive comparison of these preprocessing methods in connection to SVM is performed for a variety of benchmarks.

Alternatively, one can directly embed data in the pseudo-Euclidean vector space determined by the eigenvector decomposition of S. Pseudo-Euclidean space is a vector space equipped with a (possible indefinite) symmetric bilinear form which can be used to compute similarities and dissimilarities of data points. More precisely, a symmetric bilinear form is induced by $\langle \mathbf{x}, \mathbf{y}\rangle_{p,q} = \mathbf{x}^T I_{p,q} \mathbf{y}$ where $I_{p,q}$ is a diagonal matrix with p entries 1 and q entries -1. Taking the eigenvectors of S together with the square root of the absolute value of the eigenvalues, we obtain vectors $\mathbf{x}^i$ in pseudo-Euclidean space such that $d_{ij} = \langle \mathbf{x}^i - \mathbf{x}^j, \mathbf{x}^i - \mathbf{x}^j\rangle_{p,q}$ holds for every pair of data points. If the number of data is not limited a priori, a generalization of this concept to Krein spaces which similarly decompose into two possibly infinite dimensional Hilbert spaces is possible [25].

Vector operations can be directly transferred to pseudo-Euclidean space, i.e. we can define prototypes as linear combinations of data in this space. Hence we can perform techniques such as GLVQ explicitly in pseudo-Euclidean space since it relies on vector operations only. One problem of this explicit transfer is given by the computational complexity of the embedding which is $\mathcal{O}(N^3)$, and, further, the fact that out-of-sample extensions to new data points characterized by pairwise dissimilarities are not immediate. Because of this fact, we are interested in efficient techniques which implicitly refer to this embedding only. As a side product, such algorithms are invariant to coordinate transforms in pseudo-Euclidean space.

The key assumption is to restrict prototype positions to linear combinations of data points of the form

$$\mathbf{w}^j = \sum_i \alpha_{ji} \mathbf{x}^i \text{ with } \sum_i \alpha_{ji} = 1\,.$$

Since prototypes are located at representative points in the data space, it is a reasonable assumption to restrict prototypes to the affine subspace spanned by the given data points. In this case, dissimilarities can be computed implicitly by means of the formula

$$d(\mathbf{x}^i, \mathbf{w}^j) = [D\cdot\alpha_j]_i - \frac{1}{2}\cdot\alpha_j^T D \alpha_j \tag{6}$$

where $\alpha_j = (\alpha_{j1}, \ldots, \alpha_{jn})$ refers to the vector of coefficients describing the prototype $\mathbf{w}^j$ *implicitly*, as shown in [16]. Neither the prototypes nor the original points, related to the dissimilarity matrix, are expected to exist in a vectorial space. This observation constitutes the key to transfer GTM and GLVQ to relational data without an explicit embedding in pseudo-Euclidean space.

5. RELATIONAL GENERATIVE TOPOGRAPHIC MAPPING

GTM has been extended to general dissimilarities in [14]. We shortly recall the approach for convenience. As before, targets $\mathbf{t}^k$ in pseudo-Euclidean space induce a mixture distribution in the data space based on the dissimilarities. Targets are obtained as images of points $\mathbf{w}^k$ in latent space via a generalized linear regression model where, now, the mapping is to the coefficient vectors α which implicitly represent the targets:

$$y : \mathbf{w} \mapsto \boldsymbol{\alpha} = \Phi(\mathbf{w})\cdot\mathbf{W}$$

with images in $\mathbb{R}^N$ according to the dimensionality of the coefficients α.

The restriction

$$\sum_i [\Phi(\mathbf{w}k)\cdot\mathbf{W}]_i = \sum_i \alpha_{ki} = 1$$

is automatically fulfilled for optima of the data log likelihood. Hence the likelihood function can be computed based on (1) and the distance computation can be performed indirectly using (6). An EM optimization scheme leads to solutions for the parameters β and $\mathbf{W}$, and an expression for the hidden variables given by the responsibilities of the modes for the data points. Algorithmically, Eqn. (2) using (6) and the optimization of the expectation

$$\sum_{k,i} R_{ki}(\mathbf{W}_{\text{old}}, \beta_{\text{old}}) \ln p(\mathbf{x}^i | \mathbf{w}k, \mathbf{W}_{\text{new}}, \beta_{\text{new}})$$

with respect to $\mathbf{W}$ and β take place in turn. The latter yields model parameters which can be determined in analogy to (3,4) where, now, functions Φ map from the latent space to the space of coefficients α and $\mathbf{X}$ denotes the unity matrix in the space of coefficients. We refer to this iterative update scheme as relational GTM (RGTM). Initialization takes place by referring to the first MDS directions of $\mathbf{D}$. See [14] for details.

6. RELATIONAL LEARNING VECTOR QUANTIZATION

We use the same principle to extend GLVQ to relational data. Again, we assume a symmetric dissimilarity matrix D with zero diagonal is given. We assume that a prototype $\mathbf{w}^j$ is represented implicitly by means of the coefficient vectors α_j. Then we can use the equivalent characterization of distances (6) in the GLVQ cost function (5) leading to the costs of relational GLVQ (RGLVQ):

$$\begin{aligned}
E_{\text{RGLVQ}} &= \sum_i f\left(\frac{\xi(i)^+ - \zeta(i)^+ - \xi(i)^- + \zeta(i)^-}{\xi(i)^+ - \zeta(i)^+ + \xi(i)^- - \zeta(i)^-}\right)\\
\xi(i)^+ &= [D\alpha^+]_i\\
\xi(i)^- &= [D\alpha^-]_i\\
\zeta(i)^+ &= \frac{1}{2}\cdot(\alpha^+)^T D\alpha^+\\
\zeta(i)^- &= \frac{1}{2}\cdot(\alpha^-)^T D\alpha^-
\end{aligned}$$

where as before the closest correct and wrong prototype are referred to, corresponding to the coefficients α^+ and $\alpha-$, respectively. A simple stochastic gradient descent leads to adaptation rules for the coefficients α^+ and α^- in relational GLVQ: component k of these vectors is adapted as

$$\begin{aligned}
\Delta\alpha_k^+ &\sim \frac{-\Phi'(\mu(\mathbf{x}^i))}{(\mu^+(\mathbf{x}^i))^{-1}} \cdot \frac{\partial\left([D\alpha^+]_i - \frac{1}{2}(\alpha^+)^T D\alpha^+\right)}{\partial\alpha_k^+}\\
\Delta\alpha_k^- &\sim \frac{\Phi'(\mu(\mathbf{x}^i))}{(\mu^-(\mathbf{x}^i))^{-1}} \cdot \frac{\partial\left([D\alpha^-]_i - \frac{1}{2}(\alpha^-)^T D\alpha^-\right)}{\partial\alpha_k^-}
\end{aligned}$$

where $\mu(\mathbf{x}^i)$, $\mu^+(\mathbf{x}^i)$, and $\mu^-(\mathbf{x}^i)$ are as above. The partial derivative yields

$$\frac{\partial\left([D\alpha_j]_i - \frac{1}{2}\cdot\alpha_j^T D\alpha_j\right)}{\partial\alpha_{jk}} = d_{ik} - \sum_l d_{lk}\alpha_{jl}$$

Naturally, alternative gradient techniques such as line search can be used in a similar way.

After every adaptation step, normalization takes place to guarantee $\sum_i \alpha_{ji} = 1$. This way, a learning algorithm which adapts prototypes in a supervised manner similar to GLVQ is given for general dissimilarity data, whereby prototypes are implicitly embedded in pseudo-Euclidean space. The prototypes are initialized as random vectors, i.e we initialize α_{ij} with small random values such that the sum is one. It is possible to take class information into account by setting all α_{ij} to zero which do not correspond to the class of the prototype.

For both, RGTM and RGLVQ, out-of-sample extension of the model to new data is possible immediately. It can be based on an observation made in [16]: given a novel data point $\mathbf{x}$ characterized by its pairwise dissimilarities $D(\mathbf{x})$ to the data vectors $\mathbf{x}^i$ used for training, the dissimilarity of $\mathbf{x}$ to a prototype represented by α_j is

$$d(\mathbf{x}, \mathbf{w}^j) = D(\mathbf{x})^T \cdot \alpha_j - \frac{1}{2}\cdot\alpha_j^T D\alpha_j.$$

This can be directly used to compute responsibilities for RGTM or the closest prototype for RGLVQ, respectively.

7. THE NYSTRÖM APPROXIMATION

Both techniques, RGTM and RGLVQ depend on the full dissimilarity matrix D. This is of size N^2, hence the techniques have quadratic complexity with respect to the given number of data points. This is infeasible for large N: restrictions are given by the main memory (assuming double precision and 12 GB main memory, the limit is currently at about 30,000 data points), and the time necessary to compute dissimilarities and train the models based thereon (assuming 1ms for one dissimilarity computation, which is quite reasonable for complex dissimilarities e.g. based on alignment techniques, a matrix of less than 10,000 data points can be computed in 12 h on a dual core machine.) Therefore, approximation techniques which reduce the effort to a linear one would be very desirable.

7.1. *Nyström approximation for similarity data*

Nyström approximation technique has been proposed in the context of kernel methods in [38]. Here, we give a short review of this technique.

One well known way to approximate a $N \times N$ Gram matrix, is to use a low-rank approximation. This can be done by computing the eigendecomposition of the kernel $\mathbf{K} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{U}$ is a matrix, whose columns are orthonormal eigenvectors, and $\mathbf{\Lambda}$ is a diagonal matrix consisting of eigenvalues $\mathbf{\Lambda}_{11} \geq \mathbf{\Lambda}_{22} \geq \ldots \geq 0$, and keeping only the m eigenspaces which correspond to the m largest eigenvalues of the matrix. The approximation is $\mathbf{K} \approx \mathbf{U}_{N,m}\mathbf{\Lambda}_{m,m}\mathbf{U}_{m,N}$, where the indices refer to the size of the corresponding submatrix. The Nyström method approximates a kernel in a similar way, without computing the eigendecomposition of the whole matrix, which is an $O(N^3)$ operation.

By the Mercer theorem kernels $k(\mathbf{x}, \mathbf{y})$ can be expanded by orthonormal eigenfunctions ψ_i and non negative eigenvalues λ_i in the form

$$k(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty} \lambda_i \psi_i(\mathbf{x}) \psi_i(\mathbf{y}).$$

The eigenfunctions and eigenvalues of a kernel are defined as the solution of the integral equation

$$\int k(\mathbf{y}, \mathbf{x}) \psi_i(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \lambda_i \psi_i(\mathbf{y}),$$

where $p(\mathbf{x})$ is the probability density of $\mathbf{x}$. This integral can be approximated based on the Nyström technique by sampling $\mathbf{x}^k$ i.i.d. according to $p(\mathbf{x})$:

$$\frac{1}{m} \sum_{k=1}^{m} k(\mathbf{y}, \mathbf{x}^k) \psi_i(\mathbf{x}^k) \approx \lambda_i \psi_i(\mathbf{y}).$$

Using this approximation and the matrix eigenproblem equation

$$\mathbf{K}^{(m)}\mathbf{U}^{(m)} = \mathbf{U}^{(m)}\mathbf{\Lambda}^{(m)}$$

of the corresponding $m \times m$ Gram sub-matrix $\mathbf{K}^{(m)}$ we can derive the approximations for the eigenfunctions and eigenvalues of the kernel k

$$\lambda_i \approx \frac{\lambda_i^{(m)}}{m}, \quad \psi_i(\mathbf{y}) \approx \frac{\sqrt{m}}{\lambda_i^{(m)}} \mathbf{k}_y \mathbf{u}_i^{(m)}, \tag{7}$$

where $\mathbf{u}_i^{(m)}$ is the ith column of $\mathbf{U}^{(m)}$. Thus, we can approximate ψ_i at an arbitrary point $\mathbf{y}$ as long as we know the vector $\mathbf{k}_y = (k(\mathbf{x}^1, \mathbf{y}), ..., k(\mathbf{x}^m, \mathbf{y}))^T$.

For a given $N \times N$ Gram matrix $\mathbf{K}$ we randomly choose m rows and respective columns. The corresponding indices are also called landmarks, and should be chosen such that the data distribution is sufficiently covered. A specific analysis about selection strategies was recently discussed in [40]. We denote these rows by $\mathbf{K}_{m,N}$. Using the formulas (7) we obtain $\tilde{\mathbf{K}} = \sum_{i=1}^{m} 1/\lambda_i^{(m)} \cdot \mathbf{K}_{m,N}^T \mathbf{u}_i^{(m)} (\mathbf{u}_i^{(m)})^T \mathbf{K}_{m,N}$, where $\lambda_i^{(m)}$ and $\mathbf{u}_i^{(m)}$ correspond to the $m \times m$ eigenproblem. Thus we get, $\mathbf{K}_{m,m}^{-1}$ denoting the Moore-Penrose pseudoinverse,

$$\tilde{\mathbf{K}} = \mathbf{K}_{m,N}^t \mathbf{K}_{m,m}^{-1} \mathbf{K}_{m,N}. \tag{8}$$

as an approximation of $\mathbf{K}$. This approximation is exact, if $\mathbf{K}_{m,m}$ has the same rank as $\mathbf{K}$.

7.2. *Nyström approximation for dissimilarity data*

For dissimilarity data, a direct transfer is possible, see [13] for preliminary work on this topic. According to the spectral theorem, a symmetric dissimilarity matrix $\mathbf{D}$ can be diagonalized $\mathbf{D} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ with $\mathbf{U}$ being a unitary matrix whose column vectors are the orthonormal eigenvectors of $\mathbf{D}$ and $\mathbf{\Lambda}$ a diagonal matrix with the eigenvalues of $\mathbf{D}$, which can be negative for non-Euclidean distances. Therefore the dissimilarity matrix can be seen as an operator

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} \lambda_i \psi_i(\mathbf{x}) \psi_i(\mathbf{y})$$

where $\lambda_i \in \mathbb{R}$ correspond to the diagonal elements of $\mathbf{\Lambda}$ and ψ_i denote the eigenfunctions. The only difference to an expansion of a kernel is that the eigenvalues can be negative. All further mathematical manipulations can be applied in the same way.

Using the approximation (8) for the distance matrix, we can apply this result for RGTM. It allows to approximate dissimilarities between a prototype $\mathbf{w}^k$ represented by a coefficient vector α_k and a data point $\mathbf{x}^i$ in the way

$$\begin{aligned} d(\mathbf{x}^i, \mathbf{w}^k) \quad \approx \quad & \left[\mathbf{D}_{m,N}^T \left(\mathbf{D}_{m,m}^{-1} \left(\mathbf{D}_{m,N} \boldsymbol{\alpha}_k\right)\right)\right]_i \quad (9) \\ & -\frac{1}{2} \cdot \left(\boldsymbol{\alpha}_k^T \mathbf{D}_{m,N}^T\right) \cdot \\ & \left(\mathbf{D}_{m,m}^{-1} \left(\mathbf{D}_{m,N} \boldsymbol{\alpha}_k\right)\right) \end{aligned}$$

with a linear submatrix of m rows and a low rank matrix $\mathbf{D}_{m,m}$. Performing these matrix multiplications from right to left, this computation is $\mathcal{O}(m^2 N)$ instead of $\mathcal{O}(N^2)$, i.e. it is linear in the number of data points N, assuming fixed approximation m.

We use this approximation directly in RGTM and RGLVQ by taking a random subsample of m points to approximate the dissimilarity matrix. The percentage m is differed during training showing the effect of the approximation on the percentage used for the approximation.

A benefit of the Nyström technique is that it can be decided priorly which linear parts of the dissimilarity matrix will be used in training. Therefore, it is sufficient to compute only a linear part of the full dissimilarity matrix D to use these methods. A drawback of the Nyström approximation is that a good approximation can only be achieved if the rank of $\mathbf{D}$ is kept as much as possible, i.e. the chosen subset should be representative. We will see that the method can be used in many (though not all) settings leading to a considerable speed-up.

8. EXPERIMENTS

We evaluate the techniques on three benchmarks from the biomedical domain:

- The *Copenhagen Chromosomes data set* constitutes a benchmark from cytogenetics [22]. A set of 4,200 human chromosomes from 21 classes (the autosomal chromosomes) are represented by grey-valued images. These are transferred to strings measuring the thickness of their silhouettes. These strings are compared using edit distance with insertion/deletion costs 4.5.
- The *vibrio data set* consists of 1,100 samples of vibrio bacteria populations characterized by mass spectra. The spectra encounter approx. 42,000 mass positions. The full data set consists of 49 classes of vibrio-sub-species. The mass spectra are preprocessed with a standard workflow using the BioTyper software [23]. According to the functional form of mass spectra, dedicated similarities as provided by the BioTyper software are used [23].
- Similar to an application presented in [19], we extract roughly 11,000 protein sequences of the *SwissProt data base* according to 32 functional labels given by PROSITE [11]. Sequence alignment is done using local alignment by means of the Smith-Waterman algorithm.

We compare the two different prototype-based methods, RGTM and RGLVQ and the Nyström approximations thereof. The following parameters are used:

- Evaluation is done by means of the generalization error in a ten fold repeated cross-validation with ten repeats. (Only two-fold cross-validation and five repeats for SwissProt.) For RGTM, posterior labeling on the test set is used.
- The number of prototypes is chosen roughly following the number of priorly known classes. For RGLVQ, we use 63 prototypes for the Chromosomes data set, 49 prototypes for the Vibrio data set, and 64 prototypes for the SwissProt data set. For RGTM, to account for the topological constraints which result in prototypes outside the convex hull of the data, we use lattices of size 20×20 for Chomosomes and Vibrio and 40×40 for SwissProt. 10×10 base functions are used in both cases.
- Training takes place until convergence which is 50 epochs for the small data sets and 5 epochs for SwissProt.

- For the Nyström approximation, we report the results obtained when sampling 1% and 10% of the data.
- For the SwissProt data set, the speed up of the method can clearly be observed due to the large size of the data set. We also report the CPU time in seconds taken for one cross-validation run on a 24 Intel(R) Xeon X5690 machine with 3.47GHz processors and 48 GB DDR3 1333MHz memory. The experiments are implemented in Matlab.

The results of the techniques are collected in Tab. 1. The transformations for the SVM are done in accordance to [8] with results taken form [27] [†] Since there is not yet a best way to do this eigenvalue correction, all approaches have to be tried on the training data and the best results is chosen. Most of these transformation require a singular value decomposition of the similarity matrix, with a complexity of ($O(N^3)$). In contrast the proposed relational methods do not require any kind of preprocessing but can be applied directly on the given, symmetric, dissimilarity matrices.

For both data sets, Chromosomes and Vibrio, the classification accuracy of RGLVQ is better as compared to the unsupervised RGTM which can be attributed to the fact that the primal can take the priorly known class information into account during training. Interestingly, the results of the Nyström approximation are quite diverse. In some cases, the classification accuracy is nearly the same as for the original method, in others, the accuracy even increases when taking the Nyström approximation. In some cases, the result drops down by more than 40%. Interestingly, the result is not monotonic with respect to the size used to approximate the data, and it is also not consistent for the two algorithms. While Nyström approximation is clearly possible for RGLVQ and the Vibrio data set, the quality depends very much on the approximation parameters for RGTM.

Thus it seems that the quality of the approximation is not necessarily better the larger the fraction of the data taken for approximation, and it seems that the techniques are affected to a different degree by this approximation quality.

To shed some light on this aspect, we directly evaluate the quality of the Nyström approximation as follows: we repeatedly sample a different fraction of the data set and evaluate the distance of the approximated matrix and the original one. Since both methods do not depend on the exact size of the dissimilarities, but rather the ranking induced by the values is important, we evaluate the spearman correlation of the resulting columns. The results are depicted in Fig. 1,2. Interestingly, the resulting quality is not monotonic with respect to the size of the subsample taken for the approximation. Rather, the spearman correlation drops down for all settings and larger percentage of the subsample for all three cases. This can probably be attributed to the fact that, for larger values, noise in the data accounts for random fluctuations of the ranks rather than an approximation of the underlying order. Hence it can be advisable to test different, on particular also comparably small subsamples to arrive at a good approximation.

The speed-up of the techniques by means of the approximation has been evaluated for the SwissProt data set as a comparably large data set. Note that the current limit regarding memory restrictions for a standard memory size of 12 GB would allow at most 30,000 samples, hence the SwissProt data data also in the order of magnitude of this limit. Interestingly, the speed-up is more than 2.5 if 10% are taken and close to six if only 1% is chosen for RGLVQ. Hence the Nyström approximation can contribute to a considerable speed-up in these cases, while not deteriorating the quality for RGLVQ or RGTM.

9. CONCLUSIONS

Relational GTM offers a highly flexible tool to simultaneously cluster and order dissimilarity data in a topographic mapping. It relies on an implicit pseudo-Euclidean embedding of data such that dissimilarities become directly accessible. We have proposed a similar extension of supervised prototype based methods, more precisely GLVQ, to obtain a high quality classification scheme for dissimilarity

[†] SVM results are obtained by a standard C++ implementation, while the other experiments are done in pure matlab, hence the CPU time is not comparable here.

	Chromosome	***Vibrio***	***SwissProt***	CPU
RGTM	88.10 (0.7)	94.70 (0.5)	69.90 (2.5)	9656
RGTM (Ny 0.01)	87.80 (2.70)	54.70 (3.10)	74.40 (2.70)	786
RGTM (Ny 0.10)	51.60 (6.60)	93.90 (0.70)	82.20 (4.50)	1631
RGLVQ	92.70 (0.20)	100.00(0.00)	82.30(0.00)	24481
RGLVQ (Ny 0.01)	78.40 (0.10)	99.10(0.10)	87.00(0.00)	4179
RGLVQ (Ny 0.10)	78.20 (0.40)	99.20(0.20)	83.40(0.20)	9696
SVM*	92.50 (3.30)	100.00(0.00)	98.40(0.10)	-
SVM* **(Ny 0.01)**	95.60 (1.30)	85.27(4.32)	86.30(0.10)	-
SVM* **(Ny 0.1)**	68.80 (1.90)	99.82(0.57)	63.00(1.50)	-

Table 1: Results of the methods on the three data sets, the generalization error is reported, the standard deviation is given in parentheses. For SwissProt, we also report the CPU time for one run in seconds.

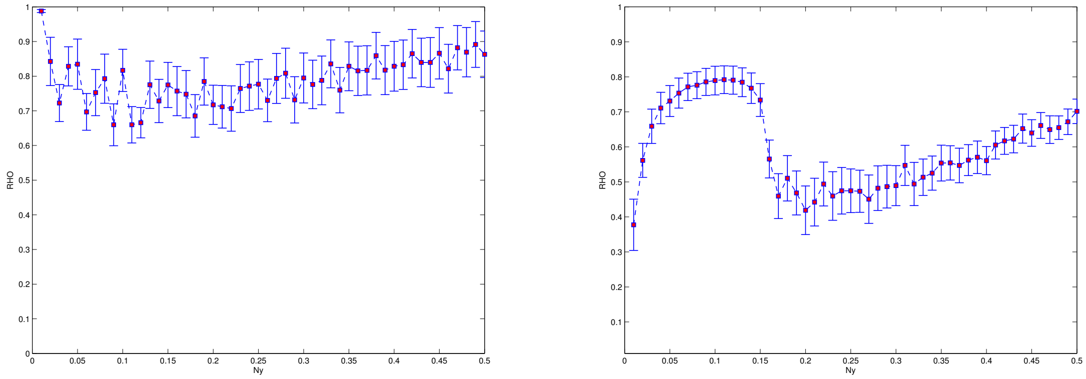

Figure 1: Quality of the Nyström approximation as evaluated by the Spearman correlation of the rows of the approximated matrix and the original one. The approximation is based on a different fraction of the data set as indicated by the x-axis. The graphs show the result for the Chromosomes (left) and Vibrio (right).

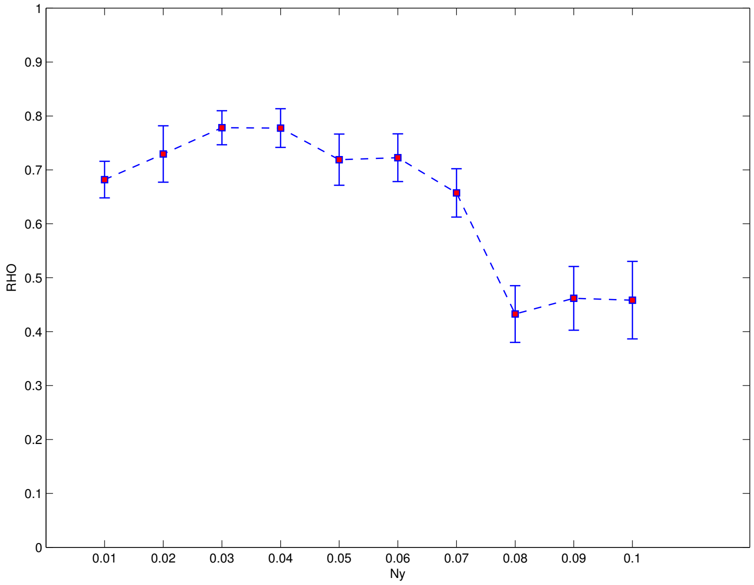

Figure 2: Quality of the Nyström approximation as evaluated by the Spearman correlation for SwissProt.

data.

Due to the dependency on the full matrix, both methods requires squared time complexity and memory to store the dissimilarities. We have proposed a speed-up techniques which leads to linear effort: the Nyström approximation. Using three examples from the biomedical domain, we demonstrated that already for comparably small data sets the technique can largely enhance speed while not loosing too much information contained in the data.

Interestingly, the quality of the Nyström technique does not scale monotonously with the sample size taken for the approximation. Rather, depending on the data characteristics, smaller samples might lead to a better job. Therefore, it is always worthwhile to test different sample sizes to achieve the optimum balance of accuracy and speed.

Acknowledgment
This work was supported by the "German Science Foundation (DFG)" under grant number HA-2719/4-1. Further, financial support from the Cluster of Excellence 277 Cognitive Interaction Technology funded in the framework of the German Excellence Initiative is gratefully acknowledged. We would like to thank Dr. Markus Kostrzewa, Dr. Thomas Maier, Dr. Stephan Klebel, Dr. Thomas Elssner and Dr. Karl-Otto Kruter (all Bruker Daltonik GmbH), for providing the Vibrio data set and additional support during pre-processing and interpretation.

1. N. Alex, A. Hasenfuss, and B. Hammer. Patch clustering for massive data sets. *Neurocomputing*, 72(7-9):1455–1469, 2009.
2. W. Arlt, M. Biehl, A.E. Taylor, S. Hahner, R. Libe, B.A. Hughes, P. Schneider, D.J. Smith, H. Stiekema, N. Krone, E. Porfiri, G. Opocher, J. Bertherat, F. Mantero, B. Allolio, M. Terzolo, P. Nightingale, C.H.L. Shackleton, X. Bertagna, M. Fassnacht, P.M. Stewart Urine Steroid Metabolomics as a Biomarker Tool for Detecting Malignancy in Adrenal Tumors *J. of Clinical Endocrinology & Metabolism*, Vol. 96, No. 12, pp. 3775-3784, 2011
3. Wesam Barbakh and Colin Fyfe. Online clustering algorithms. *Int. J. Neural Syst.*, 18(3):185–194, 2008.
4. S. B. Barbuddhe, T. Maier, G. Schwarz, M. Kostrzewa, H. Hof, E. Domann, T. Chakraborty, and T. Hain, Rapid identification and typing of listeria species by matrix-assisted laser desorption ionization-time of flight mass spectrometry, *Applied and Environmental Microbiology*, vol. 74, no. 17, pp. 5402–5407, 2008.
5. M. Biehl, A. Ghosh, and B. Hammer, Dynamics and generalization ability of LVQ algorithms, J. Machine Learning Research 8 (Feb):323-360, 2007.

6. C. Bishop, M. Svensen, and C. Williams. The generative topographic mapping. *Neural Computation* 10(1):215-234, 1998.
7. Romain Boulet, Bertrand Jouve, Fabrice Rossi and Nathalie Villa-Vialaneix. Batch kernel SOM and related Laplacian methods for social network analysis. *Neurocomputing*, 71(7-9:1257-1273, 2008.
8. Y. Chen, E. K. Garcia, M. R. Gupta, A. Rahimi, L. Cazzanti; Similarity-based Classification: Concepts and Algorithms, Journal of Machine Learning Research 10(Mar):747–776, 2009.
9. J. Shawe-Taylor and N. Cristianini, Kernel Methods for Pattern Analysis and Discovery *Cambridge University Press*, 2004
10. M. Cottrell, B. Hammer, A. Hasenfuss, and T. Villmann. Batch and median neural gas. *Neural Networks*, 19:762–771, 2006.
11. E. Gasteiger, A. Gattiker, C. Hoogland, I. Ivanyi, R.D. Appel, A. Bairoch, ExPASy: the proteomics server for in-depth protein knowledge and analysis, Nucleic Acids Res. 31:3784-3788, 2003.
12. Roberto Gil-Pita and Xin Yao. Evolving edited k-nearest neighbor classifiers. *Int. J. Neural Syst.*, 18(6):459–467, 2008.
13. A. Gisbrecht, B. Mokbel, and B. Hammer. The Nystrom approximation for relational generative topographic mappings. In *NIPS workshop on challenges of Data Visualization*, 2010.
14. A. Gisbrecht, B. Mokbel, and B. Hammer. Relational Generative Topographic Mapping. Neurocomputing 74: 1359-1371, 2011.
15. T. Graepel and K. Obermayer (1999), A stochastic self-organizing map for proximity data, *Neural Computation* 11:139-155, 1999.
16. B. Hammer and A. Hasenfuss. Topographic Mapping of Large Dissimilarity Data Sets. Neural Computation 22(9):2229-2284, 2010.
17. R. J. Hathaway and J. C. Bezdek. Nerf c-means: Non-Euclidean relational fuzzy clustering. *Pattern Recognition* 27(3):429-437, 1994.
18. T. Kohonen, editor. *Self-Organizing Maps.* Springer-Verlag New York, Inc., 3rd edition, 2001.
19. T. Kohonen and P. Somervuo (2002), How to make large self-organizing maps for nonvectorial data, *Neural Networks* 15:945-952.
20. J. Laub, V. Roth, J.M. Buhmann, K.-R. Müller. On the information and representation of non-Euclidean pairwise data. *Pattern Recognition* 39:1815-1826 2006.
21. Ezequiel López-Rubio, Rafael Marcos Luque Baena, and Enrique Domínguez. Foreground detection in video sequences with probabilistic self-organizing maps. *Int. J. Neural Syst.*, 21(3):225–246, 2011.
22. C. Lundsteen, J. Phillip, and E. Granum (1980), Quantitative analysis of 6985 digitized trypsin G-banded human metaphase chromosomes, *Clinical Genetics* 18:355-370.
23. T. Maier, S. Klebel, U. Renner, and M. Kostrzewa, Fast and reliable MALDI-TOF ms–based microorganism identification, *Nature Methods*, no. 3, 2006.
24. Britta Mersch, Tobias Glasmachers, Peter Meinicke, and Christian Igel. Evolutionary optimization of sequence kernels for detection of bacterial gene starts. *Int. J. Neural Syst.*, 17(5):369–381, 2007.
25. E. Pekalska and R.P.W. Duin The Dissimilarity Representation for Pattern Recognition. Foundations and Applications. World Scientific, Singapore, December 2005.
26. A. Sato and K. Yamada. Generalized learning vector quantization. In M. C. Mozer D. S. Touretzky and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, pages 423–9, Cambridge, MA, USA, 1996. MIT Press.
27. B. Hammer and B. Mokbel and F.-M. Schleif and X. Zhu White Box Classification of Dissimilarity Data. In E. Corchado and V. Snásel and A. Abraham and M. Wozniak and M. Graña and S.-B. Cho, editors, *Hybrid Artificial Intelligent Systems - 7th International Conference, HAIS 2012, Salamanca, Spain, March 28-30th, 2012. Proceedings, Part I*, pages 309-321, Springer.
28. B. Hammer and F.-M. Schleif and X. Zhu Relational Extensions of Learning Vector Quantization. In B.-L. Lu and L. Zhang and J. T. Kwok, editors, *Neural Information Processing - 18th International Conference, ICONIP 2011, Shanghai, China, November 13-17, 2011, Proceedings, Part II*, pages 481-489, Springer.
29. Efficient kernelized prototype based classification, Frank-M. Schleif, T. Villmann, B. Hammer, P. Schneider, M. Biehl, International Journal of Neural Systems, vol. 21, no, 6, pp. 443-457, 2011.
30. Ranking-based Kernels in Applied Biomedical Diagnostics using Support Vector Machine, V. Jumutc, P. Zayakin, A. Borisov. International Journal of Neural Systems, vol. 21, no, 6, pp. 459-473, 2011.
31. Discovering Significant Evolution Patterns from Satellite Image Time Series, F. Petitjean, F. Masseglia, P. Gancarski, , and G. Forestier International Journal of Neural Systems, vol. 21, no, 6, pp. 475-489, 2011.
32. E. Pekalska, R. P. W. Duin, S. Günter and H. Bunke On Not Making Dissimilarities Euclidean *SSPR/SPR'2004*, pp. 1145-1154, 2004
33. P. Schneider, M. Biehl, and B. Hammer, Adaptive relevance matrices in learning vector quantization,' *Neural Computation*, vol. 21, no. 12, pp. 3532–3561, 2009.
34. Principal Manifolds and Graphs in Practice: From Molecular Biology to Dynamical Systems, A.N. Gorban and A. Zinovyev International Journal of Neural

Systems, vol. 20, no, 3, pp. 219-232, 2011.

35. S. Seo and K. Obermayer (2004), Self-organizing maps and clustering methods for matrix data, *Neural Networks* 17:1211-1230.
36. S. Seo and K. Obermayer. Soft learning vector quantization. *Neural Computation*, 15(7):1589–1604, 2003.
37. L. Shi and Y. Shi and Y. Gao A Novel Approach of Clustering with an Extended Classifier System *International Journal of Neural Systems*, 21(1):79–93, 2011.
38. C. K. I. Williams, M. Seeger. Using the Nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems 13*: 682-688, 2001
39. H. Yin. On the equivalence between kernel self-organising maps and self-organising mixture density networks. *Neural Networks*, 19(6-7):780–784, 2006.
40. K. Zhang and J. T. Kwok. Clustered Nyström method for large scale manifold learning and dimension reduction *IEEE Transactions on Neural Networks*, 21(10): 1576-1587, 2010