Accelerating Kernel Clustering for Biomedical Data Analysis

Andréj Gisbrecht, Barbara Hammer, Frank-Michael Schleif, and Xibin Zhu
University of Bielefeld
CITEC Center of Excellence
D-33594 Bielefeld, Germany
Email: bhammer@techfak.uni-bielefeld.de

Abstract—The increasing size and complexity of modern data sets turns modern data mining techniques to indispensable tools when inspecting biomedical data sets. Thereby, dedicated data formats and detailed information often cause the need for problem specific similarities or dissimilarities instead of the standard Euclidean norm. Therefore, a number of clustering techniques which rely on similarities or dissimilarities only have recently been proposed. In this contribution, we review some of the most popular dissimilarity based clustering techniques and we discuss possibilities how to get around the usually squared complexity of the models due to their dependency on the full dissimilarity matrix. We evaluate the techniques on two benchmarks from the biomedical domain.

I. INTRODUCTION

The amount of electronic data available today in the context of biomedical applications increases rapidly due to improved data formats, storage possibilities, and technical developments which allow to automatically gather data in digital form. This causes the need for efficient data mining techniques which help humans to rapidly scan through the given data volumes and extract relevant information. Clustering offers one classical tool to structure large data sets. Thereby, many different clustering technologies exist such as hierarchical methods, model based techniques, graph clustering, or prototype based techniques [1].

Prototype based clustering techniques focus on specific cluster structures which are represented by prototypes and the data points which are closest to these prototypes in a given distance measure. Popular methods include, for example, classical k-means clustering, fuzzy-k-means, neural gas, self-organizing maps, or median clustering [1]. Typically, the methods aim at a minimization of the quantization error or a variant thereof, which measures the average distance of data to their respective prototypes. This representation of clusters has several benefits: clusters can be represented in an intuitive way by presenting their central representative prototype. These can often be inspected manually by experts, and the corresponding cluster structure can then be manually verified by experts. Clustering takes place in an intuitive way based on the notion of similarity to the representative prototypes. Further, the prototypes offer a compression of the given data set.

Most classical prototype-based techniques, however, have the drawback that they have been proposed for Euclidean data only. That means, data has to be embedded into a real-vector space for clustering. Novel sensor technologies, multiple modalities, and dedicated data formats often cause the need for specific structural comparisons, making an embedding in a finite dimensional Euclidean vector space problematic. Examples are bioinformatics sequences which are usually compared using sequence alignment, mass spectra which require a pairwise alignment and comparison of peaks, biological networks which rely on graphs and corresponding comparisons, or biomedical images which also require an alignment of the characteristic parts presented in the image. Hence, classical Euclidean prototype based clustering techniques are not applicable in such domains.

Recently, a variety of models and extensions has been proposed which extends prototype based clustering techniques towards general similarities or dissimilarities. That means, the clustering method does only rely on pairwise distances of data rather than explicit vectors. The main difficulty consists in the fact that the definition and location of prototypes is no longer obvious if no surrounding vector space is available where these prototypes can smoothly be adapted. Correspondingly, specific ways of how to define prototypes in an abstract space characterized by pairwise distances need to be chosen. Examples for such methods include affinity propagation (AP) [2] which constitutes an exemplar based clustering, i.e. it restricts prototype locations to data points such that the standard quantization error is still well defined. Optimization takes place by means of a factor graph representation of the quantization error and corresponding optimization using the max-sum algorithm. Relational neural gas (RNG) and kernel neural gas (KNG) rely on an only indirect representation of the prototypes by means of weighting factors for the given data [3], [4]. These can be adapted smoothly since the weighting factors are real numbers. If the weights sum up to one, an indirect computation of the distance of prototypes and data is still possible. The relational generative topographic mapping (RGTM) takes a similar perspective, but it integrates this idea into a generative statistical model which explains the data such that optimization of the parameters can be based on a maximization of the data likelihood [3].

All of these approaches have the drawback that they rely on pairwise similarities or dissimilarities of data. This matrix
scales quadratic with the number of data and, correspondingly, the clustering techniques have quadratic effort. This makes them infeasible if large size data sets are dealt with. In consequence, it is worthwhile to investigate approximative versions of the methods which allow to obtain qualitatively accurate results in less time, preferably linear time.

Recently, two different techniques which allow to achieve this goal have been proposed: patch processing addresses the data in fixed size patches only, this way visiting an only linear subpart of the full dissimilarity matrix and speeding up the clustering technique to a linear method [3]. This has been proposed in the context of RNG. In this contribution, we demonstrate the feasibility of this patch processing for RNG, and we also extend this technique to AP to a very powerful linear exemplar-base method. As an alternative, the Nyström approximation constitutes a classical method to speed up kernel techniques [5]. Here, we investigate its applicability to KNG and RGT M and we demonstrate that its suitability heavily relies on the internal properties of the dissimilarity matrix, in particular its spectrum.

The suitability of the resulting linear methods is tested on two real-life benchmarks: a medical image classification task and a classification task for vibrio bacteria characterized by mass spectra. For both data sets, dedicated problem specification techniques of machine learning, first, and we introduce extensions to similarity or dissimilarity measures as well as efficient approximations for large data sets, afterwards. Then, we compare the techniques on benchmark data from bioinformatics, and conclude with a discussion.

II. METHODS

We focus on prototype based clustering methods which represent clusters in terms of prototypical representatives.

A. Euclidean Methods

In the standard Euclidean setting, data \( \vec{v}_i \in \mathbb{R}^n \) are represented by \( K \) prototypes \( \vec{w}_j \in \mathbb{R}^n \); the receptive field \( R_j \) of a prototype \( \vec{w}_j \) consists of all data points \( \vec{v}_i \) which are closest to \( \vec{w}_j \) as measured by the Euclidean distance \( d(\vec{v}_i, \vec{w}_j) = \| \vec{v}_i - \vec{w}_j \|^2 \), breaking ties deterministically. The goal is to minimize the quantization error

\[
E_{qe} := \sum_{i,j; \vec{v}_i \in R_j} d(\vec{v}_i, \vec{w}_j).
\] (1)

There exist different classical methods which achieve this goal. We shortly describe three models in the following. Thereby, we can refer to distances such as the Euclidean distance, or similarities such as the Euclidean dot product as required, since similarities and dissimilarities can be transferred into each other using classical double centering (see e.g. [9]).

GTM: The generative topographic mapping (GTM) relies on a generative statistical model, a constraint mixture of Gaussians induced by a low dimensional lattice in latent space. The lattice points \( \vec{u}_j \) are mapped to prototypes \( \vec{w}_j = \Phi(\vec{u}_j) \cdot W \) by means of a generalized linear regression based on equally spaced Gaussians as base functions \( \Phi, W \) denoting the weights of this function. Every prototype \( \vec{w}_j \) induces a Gaussian distribution which variance is determined by the parameter \( \beta \)

\[
p(\vec{x}|\vec{w}_j, \beta) = (\beta/(2\pi))^{n/2} \exp(-\beta/2 \cdot d(\vec{x}, \vec{w}_j)^2)
\] (3)

These are combined in a mixture model with equal weights for the mixture components. Now training takes place such that the data log likelihood is optimized, this way determining optimum parameters \( W \) for the position of prototypes by means of the generalized linear regression from the latent space, and optimum variances of the Gaussians in the data space. It is possible to derive a rapidly converging expectation maximization (EM) algorithm as detailed in [6]; EM optimization in turn computes responsibilities

\[
R_{ij}(W, \beta) = p(\vec{x}_i|\vec{w}_j, \beta)/\sum_{j'} p(\vec{x}_i|\vec{w}_{j'}, \beta)
\] (4)

and it optimizes the parameters \( W \) and \( \beta \) by solving the linear equations

\[
\frac{1}{\beta_{new}} = \frac{1}{Nn} \cdot \sum_{ij} R_{ij}(W_{old}, \beta_{old}) d(\Phi(\vec{u}_j) W_{new}, \vec{v}_i)
\] (5)

where \( N \) denotes the number of data points, \( G \) is the diagonal matrix with entries \( G_{ij} = \sum_j R_{ij}(W, \beta) \), \( \Phi \) is the matrix of responsibilities, \( \Phi \) is the matrix of base functions evaluated at all lattice points, and \( V \) the matrix of data points.

NG: Neural gas (NG) extends the quantization error to incorporate data induced neighborhood cooperation:

\[
E_{NG} := \sum_{ij} h_\sigma(k_{ij}) d(\vec{v}_i, \vec{w}_j)
\] (7)

where \( h_\sigma(t) = \exp(-t/\sigma) \) exponentially scales the neighborhood range, and \( k_{ij} \) denotes the rank of prototype \( \vec{w}_j \) with respect to \( \vec{v}_i \), i.e. the number of prototypes \( \vec{w}_k \) with \( k \neq j \) which are closer to \( \vec{v}_i \) as measured by the Euclidean distance \( d \). Classical NG optimizes this objective by means of a stochastic gradient descent, thereby annealing the neighborhood range \( \sigma \) during training such that, in the limit, the standard quantization error is approximated [7]. The iterative adaptation rule is

\[
\vec{w}_j := \vec{w}_j + \eta \cdot h_\sigma(k_{ij}) (\vec{v}_i - \vec{w}_j)
\] (8)

where \( \eta \) denotes the learning rate. There exists a faster batch optimization scheme as introduced in [8] which in turn
optimizes prototype locations and assignments similar to an EM scheme, i.e. it consecutively computes
\[
\begin{align*}
determine \ k_{ij} \ & \text{based on } d(\vec{v}_i, \vec{w}_j) \\
\vec{w}_j := & \frac{\sum_i h_{\sigma}(k_{ij})\vec{v}_i}{\sum_i h_{\sigma}(k_{ij})}
\end{align*}
\] (9) (10)

**AP:** Affinity propagation (AP) constitutes an exemplar based clustering method, i.e. prototype locations are restricted to data points \(\vec{w}_j \in \{\vec{v}_i \mid i = 1, \ldots, N\}\). Further, it deals with similarities \(s(\vec{w}_j, \vec{v}_i)\) rather than dissimilarities such as the Euclidean dot product, for example. Obviously, the quantization error can be formulated accordingly. Since prototypes are located at discrete positions, the quantization error can no longer be optimized numerically. It is slightly reformulated as
\[
E_{\text{qe-ap}} := \sum_i s(\vec{v}_i, \vec{w}_I(i)) + \sum_i \delta_i(I)
\] (11)

where the assignment function \(I\) assigns a prototype given by an exemplar \(\vec{v}_I(i)\) to every data point \(\vec{v}_i\), and \(\delta_i(I)\) punishes invalid assignments, i.e. situations where \(I(j) \neq i\). This leads to the punishment \(\infty\) (the error should be maximized since we deal with similarities). Otherwise, \(\delta_i(I)\) is 0. Note that, this way, it is no longer necessary to specify the number of clusters, rather the number is determined by the self-similarities of the data \(s(\vec{v}_i, \vec{v}_i)\) which indicate in how far the point \(\vec{v}_i\) is available as a prototype. Typically, these self-similarities are set to the medium of the given similarities.

To numerically solve this optimization problem, the function \(E_{\text{qe-ap}}\) can be reformulated as a factor graph which can approximately be optimized by means of the max-sum algorithm. Assuming pairwise similarities \(s_{ij}\) of the data points, in turn, responsibilities
\[
r_{ij} = s_{ij} - \max_{j' \neq j} (a_{ij'} + s_{ij'})
\]
and availabilities
\[
a_{ij} = \min\{0, r_{jj} + \sum_{j' \neq i} \max\{0, r_{ij'}\}\}
\]
\[
a_{ii} = \sum_{j' \neq i} \max\{0, r_{ij'}\}
\]
are determined leading to assignments \(I(i) = \arg\max_j (a_{ij} + r_{ij})\) as detailed in [2].

**B. Extensions to (Dis-)Similarities**

With the exception of AP, all clustering techniques as introduced above have been introduced for the Euclidean setting only, since they rely on the fact that a smooth adaptation of prototypes in a Euclidean vector space is possible. Note that AP, by restricting prototype positions to exemplars, is defined for every possibly non-Euclidean similarity measure. The factor graph optimization as well as its theoretical background hold for every such setting. This property is very interesting since, often, data are not given in vectorial form, rather, pairwise similarities \(s_{ij}\) or dissimilarities \(d_{ij}\) of data points numbered \(i\) and \(j\) are available only, such as biological sequences and alignment distances.

NG and GTM cannot directly be applied in such settings. However, recently, extensions of these technologies have been proposed which can take into account a general kernel matrix or even a non-Euclidean dissimilarity matrix. Similarity and dissimilarity matrices can be transferred to each other using double centering, as already mentioned above. Because of this fact, we will not differentiate these two notations and use the matrix as required by the algorithm. Note, however, that general similarities/dissimilarities need not necessarily stem from a Euclidean vector space and negative eigenvalues might occur. In all cases, an application of the algorithms and an embedding in pseudo-Euclidean space is still possible, but mathematical guarantees which are valid in the Euclidean space do possibly not transfer to the setting (see e.g. [3]). However, as pointed out in [3], a correction of the negative eigenvalues might lead to a sever loss of information, such that the direct application of the techniques to the given similarity/dissimilarity should be preferred.

**Kernel Neural Gas:** Kernel neural gas (KNG) as introduced in [4] assumes the availability of a similarity matrix or Gram matrix \(S\) with entries \(s_{ij}\) characterizing the similarity of points numbered \(i\) and \(j\). This should be positive semidefinite to allow an interpretation by means of an embedding in an appropriate Hilbert space, i.e. \(s_{ij} = \Phi(\vec{v}_i )^T \Phi(\vec{v}_j)\) for some feature function \(\Phi\). It can, however, algorithmically be applied to general similarity matrices, as we will do in the experiments. The key idea is to identify prototypes with linear combinations in the high dimensional feature space
\[
\vec{w}_j = \sum_l \alpha_{jl} \Phi(\vec{v}_l) 
\] (12)

Then, squared distances can be computed based on the Gram matrix as follows:
\[
d(\Phi(\vec{v}_i), \vec{w}_j) = s_{ii} - 2\sum_l \alpha_{jl} s_{il} + \sum_{l,l'} \alpha_{jl} \alpha_{jl'} s_{ll'} 
\] (13)

Further, substituting the prototypes by linear combinations, the adaptation rule (8) becomes
\[
\alpha_{jl} := \alpha_{jl} (1 - \eta h_{\sigma}(k_{ij})) \text{ if } l \neq i 
\] (14)
\[
\alpha_{ji} := \alpha_{ji} (1 - \eta h_{\sigma}(k_{ij})) + \eta h_{\sigma}(k_{ij}) 
\] (15)

Note that this formula can be applied also if \(s_{ij}\) does not correspond to a Gram matrix, i.e. negative eigenvalues occur. In this case, however, an interpretation by means of an embedding space is less clear and convergence of the algorithm is not guaranteed. For a Gram matrix, we can w.l.o.g. restrict the coefficients such that prototypes are contained in the convex hull of the data points in the feature space, i.e. the coefficients are nonnegative and sum up to one.

**Relational Neural Gas:** Relational neural gas (RNG) as introduced in [3] assumes that a symmetric dissimilarity matrix \(D\) with entries \(d_{ij}\) describing pairwise dissimilarities of data is available. In principle, it is very similar to KNG with respect to the general idea. There are two differences: RNG is based on dissimilarities rather than similarities, and it solves the resulting cost function using a batch optimization with
obtained by means of this bilinear form. As before, prototypes i.e. a real vector space and a symmetric bilinear form (with probably negative eigenvalues) such that the dissimilarities are obtained by means of this bilinear form. As before, prototypes are restricted to linear combinations

\[ \tilde{u}_j = \sum_i \alpha_{jl} \Phi(\vec{v}_i) \quad \text{with} \quad \sum_l \alpha_{jl} = 1 \]  

(16)

We can put the restriction that the sum always leads to one since optimum prototypes as computed with batch NG always fulfill this restriction. Under this constraint, one can see that dissimilarities can be computed by means of the formula

\[ d(\Phi(\vec{v}_i), \tilde{u}_j) = [D^l \alpha_j]_i - \frac{1}{2} \cdot \alpha^l_j \]  

(17)

where \([\cdot]_i\) refers to component \(i\) of the vector. This allows a direct transfer of batch NG to general dissimilarities by the following iterations derived from (10)

\[ \text{determine } k_{ij} \text{ based on } d(\Phi(\vec{v}_i), \tilde{u}_j) \]

(18)

\[ \alpha_{jl} := \sum_i h_\sigma(k_{ij}) / \sum_i h_\sigma(k_{ij}) \]  

(19)

This algorithm can be interpreted as neural gas in pseudo Euclidean space for every symmetric dissimilarity matrix \(D\). If negative eigenvalues are present, however, convergence is not always guaranteed, albeit it can mostly be observed in practice [3].

Relational Generative Topographic Mapping: Relational GTM (RGTM) relies on the same principle as RNG: prototypes are restricted to linear combinations of data points

\[ \tilde{u}_j = \sum_i \alpha_{jl} \Phi(\vec{v}_i) \quad \text{with} \quad \sum_l \alpha_{jl} = 1 \]  

(20)

and, thus, an indirect smooth adaptation of prototypes becomes possible [10]. This enables a computation of distances as in (17). Prototypes are obtained as images of points in latent space. We can directly define the generalized regression model from the latent space to the space of coefficients \(\alpha_k = \Phi(\vec{u}_k) \cdot W\) where the restriction \(\sum_l [\Phi(\vec{u}_k) \cdot W]_l = 1\) is enforced. Using Lagrange optimization, it can be seen that this restriction is automatically fulfilled for the generic optima obtained by means of EM optimization. Hence, the EM scheme of GTM can directly be transferred to RGTM, relying on the equivalent expression of the distances and the direct computation of the coefficients, taking the data matrix \(V\) as identity in the space of coefficients.

C. Efficient Approximations

These choices result in generalization of classical clustering algorithms to general similarities or dissimilarities, respectively. Note that AP can directly be used for arbitrary similarities due to its restriction to exemplars. By means of the dissimilarity matrix, a very general interface to complex data formats which can involve structures and corresponding structural comparisons is offered. Note that, by means of double centering [9], similarities and dissimilarities can be converted into each other.

The methods, however, have a principled drawback: since they rely on the full similarity or dissimilarity matrix, respectively, their effort is quadratic with respect to the number of data. Even more severely, the full quadratic dissimilarity matrix has to be available to apply these methods, which can include a considerable effort for complex metrics such as e.g. alignment.

These facts make the methods unsuitable if large data sets have to be dealt with. Therefore, we discuss two different possibilities to speed up the techniques towards linear methods: the Nyström approximation and patch processing, respectively. Note that it is in general no option to just embed the data into a low dimensional vector space, since this procedure is likely prone to loss of information as discussed e.g. in [3]. Therefore, we focus on methods which directly deal with the similarity/dissimilarity matrix.

Nyström Approximation: The Nyström approximation as presented in [5] substitutes a given full Gram matrix \(S\) by a low rank approximation such that, often, linear techniques result when this alternative matrix representation is integrated into the model. As shown in [11], this principle can be generalized to dissimilarities. For a given kernel \(s\), by the Mercer theorem, one can find an expansion of the form \(s(\vec{w}, \vec{v}) = \sum_{i=1}^\infty \lambda_i \Phi_i(\vec{w}) \Phi_i(\vec{v})\) with eigenfunctions \(\Phi_i\) and eigenvalues \(\lambda_i\). These eigenvalues and eigenfunctions are the solutions of the integral equation \(\int s(\vec{w}, \vec{v}) \Phi_i(\vec{v}) d\vec{v} = \lambda_i \Phi_i(\vec{w})\), where one can approximate the integral based on the Nyström technique by sampling according to \(p\):

\[ \frac{1}{m} \cdot \sum_k s(\vec{w}, \vec{v}_k) \Phi_i(\vec{v}_k) \approx \lambda_i \Phi_i(\vec{w}) \]  

(21)

Using the matrix eigenproblem \(S^{(m)} U^{(m)} = U^{(m)} \Lambda^{(m)}\) of the \(m \times m\) Gram matrix \(S^{(m)}\), approximations for the eigenfunctions can be derived:

\[ \lambda_i \approx \lambda^{(m)}_i / m \]

(22)

\[ \Phi^{(m)}_i(\vec{w}) \approx \sqrt{m} \cdot \vec{s}_\vec{w} u^{(m)}_i \]  

(23)

where \(u^{(m)}_i\) is the \(i\)th column of \(U^{(m)}\), and \(\vec{s}_\vec{w}\) refers to the vector \(s(\vec{v}_1, \vec{w}), \ldots, s(\vec{v}_m, \vec{w})\). Thus, for a given \(N \times N\) Gram matrix, \(m\) rows and respective columns are picked randomly and the enumeration is changed such that these are the first \(m\) rows and columns. Rows are denoted by \(S_{N,m}\) and columns by \(S_{N,m'}\). Hence, using the approximation (23) we obtain \(\tilde{S} = \sum_{i=1}^m 1/\lambda^{(m)}_i \cdot S_{N,m} \cdot u^{(m)}_i \cdot u^{(m)}_i dS_{m',m}\), where \(\lambda^{(m)}_i\) and \(u^{(m)}_i\) correspond to the \(m \times m\) eigenproblem. Hence we obtain, \(S_{m',m}^{-1}\) denoting the pseudoinverse

\[ \tilde{S} = S_{N,m} S_{m,m'}^{-1} S_{m,N} \]  

(24)

which is of complexity \(O(m^3 N)\) instead of \(O(N^2)\), i.e. it is linear if the approximation quality \(m\) is fixed.
Similarly, dissimilarities $D$ can be approximated if $D$ is symmetric. Being a normal matrix, $D$ allows a diagonalization, i.e. it can be interpreted as operator $d(\vec{v}, \vec{w}) = \sum_i \lambda_i \Phi_i(\vec{v}) \Phi_i(\vec{w})$. Therefore, the same mathematical treatment as for kernels is possible, the only difference being that eigenvalues are allowed to be negative.

For GTM, this yields the approximation of the distance computation (17)

$$d(\vec{v}_i, \vec{w}_j) \approx [D_{N,m}(D_{m,m}^{-1}(D_{m,N} \alpha_j))]_i$$

which is $O(m^3N)$. Again, the approximation is exact if the number of samples $m$ is chosen according to the rank of $D$. If a subsample is chosen, bounds on the quality of the Nyström approximation for kernels can be derived as presented e.g. in [12]. The chosen samples should be representative for the full data set to obtain good approximations. Thus, the Nyström method is not appropriate if data display a trend, or a good coverage of all data by few samples is not possible.

**Patch Processing:** Patch processing takes a different perspective and processes data consecutively in patches of small size $m$. All already visited data are compressed by means of the prototypes as found by the data analysis method. These serve as additional inputs to the next patch, counted with their respective multiplicities as given by the size of their receptive fields. For this purpose, an extension of the clustering method which takes into account multiple points is necessary. Since all data are taken into account this way, processing of data sets which are not i.i.d. is possible.

This procedure has been proposed for NG in [13], it has been generalized to RNG in [3]. In both cases an extension towards multiple points is immediate. In the latter case, however, due to the implicit representation of prototypes in terms of vectors of coefficients $\alpha$, a further approximation becomes necessary: it is not feasible to represent the prototypes taking into account all data (which would yield to $O(N)$ space complexity) and the necessity to access dissimilarities of all data pairs (which would cause time complexity $O(N^2)$). Therefore, an approximation of the prototypes by their respective $k$ largest entries of the vectors, $k$ being a fixed number, is necessary.

Details about the $k$-approximation can be found in [3]. For both settings, an algorithm with time complexity $O(Nm^2)$ results.

Here we also consider the possibility to extend AP towards patch clustering, resulting in patch affinity propagation (PAP). Since AP constitutes an exemplar based clustering method, the transfer of the meta algorithm to AP is immediate. It is denoted in Fig. 1. To apply PAP, however, we need to extend AP such that it can deal with multiple data points. A direct extension of AP such that the update formulas correspond to the formulas if identical points are contained multiple times in the data set does, unfortunately, not work: exactly identical points prevent convergence of AP. Therefore, we use a simple observation to achieve this extension: if data point $\vec{v}_i$ is contained in the data set $m_i$ times, the cost function becomes

$$E_{qe} = \sum_{ij; \vec{v}_i \in R_j} m_i \cdot s(\vec{v}_i, \vec{w}_j)^2$$

(26)

Obviously, this cost function is obtained if the original similarities are substituted by $m_i \cdot s(\vec{v}_i, \vec{w}_j)$. Hence we simple use these products of similarities and multiplicities instead of the simple similarities in the update equations for the responsibilities as computed in AP. The initialization of the diagonal terms should also be adapted accordingly, putting a bias towards points with large multiplicities. In our case (where similarities are negative numbers), we achieve this by a division by the multiplicities.

**III. Experiments**

We test the algorithms for two data sets originate from biomedical applications. In both cases, data are labeled such that we can evaluate the performance of the algorithms in terms of the classification error. In both cases, we report the results of a ten fold cross-validation with 10 repeats. We compare the following methods as prototypical clustering
tools for similarity/dissimilarity data, and typical speedup techniques:

- Affinity Propagation (AP) where the self-similarities are set by binary search such that a fixed number of exemplars is appropriately obtained, starting from the median. (We accept partial deviations thereof, an exact fit of a given number of clusters often requiring additional trials.) Adaptation is done until convergence is observed.
- Patch Affinity Propagation (PAP) with patch size 100 and the same strategy to determine the self-similarities as AP.
- Relational Neural Gas (RNG) with a fixed number of neurons $K$, and an exponential annealing schedule for the neighborhood range starting from $K/2$. 100 epochs are used for training.
- Patch Relational Neural Gas (PRNG) with the same parameters as RNG for each patch, and patch size 100. The number of coefficients used for the k-approximation of the prototype vectors is set to $k = 10$.
- For the Kernel Neural Gas (KNG), a fixed number of neurons $K$ is chosen and the neighborhood is annealed in the same way as for RNG. The step size of the stochastic gradient descent is 0.01. 100 epochs are considered.
- KNG is also performed using the Nyström approximation where a specified percentage of random points is taken for the approximation.
- The generative Relational Topographic Mapping (RGTM) is done using default initialization and bandwidths. The lattice size $K$, i.e. number of prototypes, and the number of base functions $F$ is picked according to the data set.
- RGTM is also used including the Nyström approximation based on a certain percentage of the data and the same parameters as above.

As some of the methods require similarities or dissimilarities, respectively, we use the standard double-centering for the transfer of similarities and dissimilarities.

**Chromosome Images:** The Copenhagen Chromosomes data set constitutes a benchmark from cytogenetics[14]. A set of 4,200 human chromosomes from 22 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 [15].

The results of the methods as specified above are reported in Tab. I. The standard deviation is shown in parentheses. The parameters not yet specified (number of classes and number of base functions, respectively) are shown in the table. The relational neural gas algorithm achieves a cross-validation classification accuracy of more than 90%. Kernel neural gas, which is similar but based on similarities instead of dissimilarities and a stochastic gradient descent instead of a batch optimization scheme, achieves a slightly worse result. Interestingly, the effect of a Nyström approximation using only 10% of the data has an only mild influence on the result, albeit the spectrum of the similarity matrix consists of many comparably large eigenvalues, making it impossible to find a good low rank approximation of the similarity matrix (see Fig. 2). Interestingly, while RGTM yields comparably good results, it suffers a lot from the information loss due to the Nyström approximation. Due to its topological restrictions, the quality decreases by more than 30% in the same setting. Affinity propagation obtains results which are comparable to RGTM. Again, an approximation in terms of patch processing leads to a loss of the quality, albeit the effect is less pronounced than the Nyström approximation for KNG. Patch processing for RNG works very well, leading to a loss of only about 3% classification accuracy.

As a consequence, all original methods seem suited for the task, where RGTM, due to its topological constraints, does not capture the class structures in the same way as methods which are not restricted by their topology. Patch
processing or Nyström approximation lead to the loss of some information, whereby the quality severely depends on the considered method. If the method works with not many constraints (such as RNG of KNG for which no topology is fixed, nor the position of prototypes is restricted), an approximation by means of the two techniques seems feasible.

**Vibrio Bacteria:** 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, see 4 for an example. 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 [16]. As usual, mass spectra display strong functional characteristics due to the dependency of subsequent masses, such that problem adapted similarities such as described in [17], [16] are beneficial. In our case, similarities are calculated using a specific similarity measure as provided by the BioTyper software [16].

The results of the different methods obtained on this data set are displayed in Tab. II. Interestingly, AP obtains a nearly perfect classification accuracy of the data set, and this property is preserved when using the linear approximation provided by PAP. In contrast, the prototype based algorithms KNG, RNG, and RGTM obtain only about 80-84% classification accuracy. In this case, the Nyström approximation does not decrease the accuracy, thus it offers a feasible linear approximation of the method. This is mirrored in the spectrum of the vibrio data set (see Fig. 3), where a comparably small number of large eigenvalues exists, such that the approximation by only few samples does not lead to severe information loss. Correspondingly, also an approximation using only 2% still yields good results. Similar characteristics hold for patch approximations which do hardly decrease the performance of the respective original methods. Unlike for the Chromosomes data set, approximations are equally good in this case, which can be accounted for by the fact that information compression is possible for the vibrio data set due to the small number of eigenvalues with large absolute value.

**IV. Conclusions**

We have presented an overview about recent prototype based clustering techniques for dissimilarity or similarity data, including affinity propagation, neural gas, and generative topographic mapping. Being based on a square matrix, the original methods display quadratic time complexity, and, thus, they are infeasible for large data sets. We have introduced two approximation techniques: the Nyström approximation, which offers a low rank approximation of the similarity or dissimilarity matrix and often can be integrated into the models such that linear methods result, on the one hand; on the other hand, patch processing which relies on the principle to process data in patches and consecutively compress the seen data in terms of prototypes.

Both approximation methods result in linear effort with respect to the number of prototypes and, even more relevant in some cases, they rely on an only linear subpart of the full similarity or dissimilarity matrix. Obviously, since the important information can be contained in exactly the subparts not computed by the technique, worst case approximation guarantees of the methods cannot exist without further assumptions on the data. For i.i.d. sampling and patch processing, or Nyström approximations which relate to the full rank to the matrix, guarantees can be derived as discussed e.g. in [13], [12].

We have demonstrated the suitability of the approach in two benchmarks from bioinformatics. Interestingly, good approximations can be found in both cases, but the Chromosomes data set, which displays a spectrum with a large number of large values, seems more sensitive to the exact algorithmic choice. This can be accounted to the fact that for this case, information loss is more severe than for the vibrio data set where only few large size eigenvalues can be observed.

**Acknowledgment**

This work has been supported by the DFG under grant number HA2719/4-1 and by the Cluster of Excellence 277 Cognitive Interaction Technology funded in the framework of the German Excellence Initiative.

**References**

Fig. 4. Example spectrum of the vibrio data set reporting the relative mass versus the measured peak intensity


