Topographic mapping of dissimilarity data

Barbara Hammer [1], Andrej Gisbrecht[1], Alexander Hasenfuss[2], Bassam Mokbel[1], Frank-Michael Schleif[1], and Xibin Zhu[1]

[1] CITEC centre of excellence, Bielefeld University, Germany
[2] Computing Centre, TU Clausthal, Germany

Abstract. Topographic mapping offers a very flexible tool to inspect large quantities of high-dimensional data in an intuitive way. Often, electronic data are inherently non Euclidean and modern data formats are connected to dedicated non-Euclidean dissimilarity measures for which classical topographic mapping cannot be used. We give an overview about extensions of topographic mapping to general dissimilarities by means of median or relational extensions. Further, we discuss efficient approximations to avoid the usually squared time complexity.

1 Introduction

With electronic data sets increasing rapidly with respect to size and dimensionality, Kohonen's ingenious self organizing map (SOM) has lost none of its attractiveness as an intuitive data inspection tool: it allows humans to rapidly access large volumes of high dimensional data [18]. Apart from its very simple and intuitive training technique, the SOM offers a large flexibility by providing simultaneous visualization and clustering based on the topographic map formation. In consequence, application scenarios range from robotics and telecommunication up to web- and music-mining; further, the self-organizing map is a widely used technique in the emerging field of visual analytics because of its efficient and robust way to deal with large, high-dimensional data sets [17].

The classical SOM and counterparts derived from similar mathematical objectives such as the generative topographic mapping or neural gas [21, 3] have been proposed to process Euclidean vectors in a fixed feature vector space. Often, electronic data have a dedicated format which cannot easily be converted to standard Euclidean feature vectors: biomedical data bases, for example, store biological sequence data, biological networks, scientific texts, textual experiment descriptions, functional data such as spectra, data incorporating temporal dependencies such as EEG, etc. It is not possible to represent such entries by means of conventional feature vectors without loss of information, many data being inherently discrete or compositional. Rather, experts access such data by means of dedicated comparison measures such as BLAST or FASTA for biological sequences, alignment techniques for biological networks, dynamic time warping for time series, etc. From an abstract point of view, dissimilarity measures or kernels which are suited for the pairwise comparison of abstract data types such as strings, trees, graphs, or functions are used.

Already almost 10 years ago, Kohonen proposed a very intuitive way to extend SOMs to discrete data characterized by dissimilarities only [19]: instead of mean prototype positions in a Euclidean vector space, neuron locations are

restricted to data positions. The generalized median serves as a computational vehicle to adapt such restricted neurons according to given dissimilarity data. This principle can be extended to alternatives such as neural gas, and it can be substantiated by a mathematical derivative from a cost function such that convergence of the technique can be proved [7]. Depending on the characteristics of the data set, however, the positional restrictions can lead to a much worse representation of the data as compared to the capabilities of continuous updates which are possible in a Euclidean vector space.

As an alternative, specific dissimilarity measures can be linked to a nonlinear kernel mapping. Kernel versions of SOM have been proposed for example in the contribution [28] for online updates and [4] for batch adaptation; in both cases, the standard SOM adaptation which takes place in the high-dimensional feature space is done implicitly based on the kernel. Kernelization of SOM allows a smooth prototype adaptation in the feature space, but it has the drawback that it is often not applicable since many classical dissimilarity measures cannot be linked to a kernel. For such cases, so-called relational approaches offer an alternative [13]: prototypes are represented implicitly by means of a weighting scheme, and adaptation takes place based on pairwise dissimilarities of the data only. This principle has already been used in the context of fuzzy clustering [15]; in the past years, it has been successfully integrated into topographic maps such as SOM, neural gas, or the generative topographic mapping [13, 12].

Both principles, median extensions of SOM or relational versions, have the drawback of squared time complexity due to their dependency on the full dissimilarity matrix. Since the computational costs of specialized dissimilarities such as alignment for strings or trees can be quite time consuming, the main computational bottleneck of the techniques is often given by the computation of the full dissimilarity matrix. For this reason, different approximation techniques have recently been proposed which rely on only a linear subset of the full dissimilarity matrix and which reduce the computational effort to an only linear one. Two particularly promising techniques are offered by the Nyström approximation, on the one hand, which can be transferred to dissimilarities as shown in [11]. On the other hand, if a computation of the dissimilarities can be done online, patch processing offers a very intuitive and easily parallelizable scheme which can even deal with non i.i.d. data distributions [1]. This way, efficient linear time processing schemes for topographic mapping of dissimilarity data arises.

In this contribution, we define topographic mapping based on cost functions first. Afterwards, we introduce two different principles to extend the techniques to dissimilarity data: median and relational clustering. Both methods can be substantiated by mathematical counterparts linking it to cost functions and pseudo-Euclidean space, respectively. We conclude with technologies which allow to speed the topographic mapping up to linear time complexity.

2 Topographic mapping

Prototype based approaches represent data vectors $\boldsymbol{x} \in \mathbb{R}^n$ by means of prototypes $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N \in \mathbb{R}^n$ based on the standard squared Euclidean distance

$$d(\boldsymbol{x}, \boldsymbol{w}_i) = \|\boldsymbol{x} - \boldsymbol{w}_i\|^2 \tag{1}$$

The receptive field of prototype $\boldsymbol{w}_i$ is determined by the characteristic function

$$\chi_i(\boldsymbol{x}) = \begin{cases} 1 & \text{if } d(\boldsymbol{x}, \boldsymbol{w}_i) \leq d(\boldsymbol{x}, \boldsymbol{w}_j) \text{ for all } j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Given a finite set of data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$, the quantization error

$$E_{\text{qe}} = \frac{1}{2} \sum_{i,j} \chi_i(\boldsymbol{x}_j) d(\boldsymbol{x}_j, \boldsymbol{w}_i) \tag{3}$$

offers one quality measure for a prototype-based representation of data. Popular learning schemes such as k-means clustering or vector quantization are directly based on this cost term, which is optimized by means of an online gradient technique (vector quantization), or a batch approach (k-means), respectively [8]. The cost function can be interpreted as a limit case of statistical data modeling by means of a mixture of Gaussians where the centers are located at prototype positions. Batch learning results as a limit case of an EM optimization scheme of the data log likelihood in this setting [2].

Albeit the quantization error constitutes one of the most popular measures to evaluate unsupervised clustering, it is often not sufficient in practical applications due to several aspects: it suffers from numerical problems due to the multimodality of the cost function and its sensitivity to noise and outliers. In addition, further functionalities are often required in application scenarios such as the possibility to visualize the prototypes and to inspect relations in between prototypes. Both problems are addressed by topographic mapping.

Topographic mapping integrates a neighborhood structure of the prototypes into the model. This way it achieves both, a better robustness with respect to local optima, outliers, and noise in the data as well as enhanced functionality due to the explicit neighborhood relations of the prototypes. In essence, topographic mapping takes place by matching a neighborhood topology of the prototypes and the topology which is inherent in the data distribution; as a consequence, the prototypes together with its neighborhood structure can be interpreted as compressed representation of the data set and its topological structure. Concrete topographic mapping technologies differ in the way how the neighborhood structure is defined.

Neural gas (NG) as proposed by Martinetz relies on a data optimum topology which is inferred directly from the data [21]. The popular SOM imposes a fixed predefined neighborhood defined by a regular lattice topology, typically a two dimensional lattice in Euclidean or hyperbolic space [18, 24]. This way, not only a neighborhood structure is inferred but it can also directly be visualized on the computer screen. Since data and lattice topology need not coincide, topological mismatches can occur unlike in NG. The original SOM does not possess a cost function in the continuous case and its mathematical investigation is quite demanding, see e.g. [16, 18, 6]. A slight variation of the definition of the receptive fields as compared to (2), however, enables the derivation from a cost function very similar to (3)

$$E_{\text{SOM}} = \frac{1}{2} \sum_{i,j} \chi_i^*(\boldsymbol{x}_j) \sum_k \exp(-\text{nd}(i,k)/\sigma^2) d(\boldsymbol{x}_j, \boldsymbol{w}_k) \tag{4}$$

where $\text{nd}(i,j)$ refers to a priorly fixed neighborhood structure of the prototypes, e.g. their distance in a predefined two dimensional lattices, and the characteristic

function of the receptive fields $\chi_i^*(\boldsymbol{x}_j)$, unlike (2), is measured via the averaged distances $\sum_k \exp(-\mathrm{nd}(i,k)/\sigma^2) d(\boldsymbol{x}_j, \boldsymbol{w}_k)$. Online adaptation iteratively adapts the winning prototype and its neighborhood towards a given data point, while batch adaptation iterates the following two computations

$$\text{compute } \chi_i^*(\boldsymbol{x}_j) \text{ for all } i, \tag{5}$$

$$\text{adapt } \boldsymbol{w}_k := \frac{\sum_{i,j} \chi_i^*(\boldsymbol{x}_j) \cdot \exp(-\mathrm{nd}(i,k)) \cdot \boldsymbol{x}_j}{\sum_{i,j} \chi_i^*(\boldsymbol{x}_j) \cdot \exp(-\mathrm{nd}(i,k))} \tag{6}$$

It has been shown in [5] that this procedure converges in a finite number of steps towards a local optimum of the cost function. The convergence is very fast such that a good initialization is necessary to avoid topological mismatches as pointed out in [9]. For this reason, typically, an initialization by means of the two main principal components takes place, and the neighborhood σ is annealed carefully during training.

The generative topographic mapping (GTM) can be seen as a statistical counterpart of SOM which models data by a constraint mixture of Gaussians [3]. The centers are induced by lattice positions in a low dimensional latent space and mapped to the feature space by means of a smooth function, usually a generalized linear regression model. That means, prototypes are obtained as images of lattice points $\boldsymbol{v}_i$ in a two dimensional space $\boldsymbol{w}_i = f(\boldsymbol{v}_i) = \Phi(\boldsymbol{v}_i) \cdot \boldsymbol{W}$ with a matrix of fixed base functions Φ such as equally spaced Gaussians in two dimensions and a parameter matrix $\boldsymbol{W}$. Every prototype induces an isotropic Gaussian probability with variance β^{-1} which are combined in a mixture model using uniform prior over the modes. For training, the data log likelihood $\sum_j \ln \frac{1}{N} \cdot \sum_i \left(\frac{\beta}{2\pi}\right)^{n/2} \exp\left(-\frac{\beta}{2} d(\boldsymbol{x}_j, \boldsymbol{w}_i)\right)$ is optimized by means of an EM approach which yields to linear algebraic equations to determine the parameters $\boldsymbol{W}$ and β. As SOM, GTM requires a good initialization which is typically done by aligning the principal components of the data with the initial images of the lattice points. The smoothness of the mapping f, i.e. the number of base functions in Φ, determines the stiffness of the resulting topological mapping. Unlike SOM which focusses on the quantization error in the limit of small neighborhood size, this stiffness accounts for a usually better visualization behavior of GTM, see e.g. Fig. 1. It can clearly be seen that GTM respects the overall shape of the data manifold while SOM pushes prototypes towards data centers, leading to local distortions. A better preservation of the manifold shape can also be obtained using VisSOM instead of SOM [27], albeit this technique is not substantiated by a global cost function such as GTM.

3 Median clustering

Often, data are not given as vectors, rather pairwise dissimilarities $d_{ij} = d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ of data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are available. Thereby, the dissimilarity need not correspond to the Eulidean metric, and it is not clear whether data $\boldsymbol{x}_i$ can be represented as finite dimensional vectors at all. In the following, we refer to the dissimilarity matrix with entries d_{ij} as D. We assume that D has zero diagonal and that D is symmetric.

This situation causes problems for classical topographic mapping since a continuous adaptation of prototypes is no longer possible like in the Euclidean case.

One solution has been proposed in [19]: prototype locations are restricted to the positions offered by data points, i.e. we enforce $\boldsymbol{w}_i \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$. In [19] a very intuitive heuristic how to determine prototype positions in this setting has been proposed based on the generalized median. As pointed out in [7], it is possible to derive a similar learning rule from the cost function of SOM (4): Like in batch SOM, optimization takes place iteratively with respect to the assignments of data to prototypes (5) and with respect to the prototype positions. The latter step does not allow an explicit algebraic formulation such as (6) because of the restriction of prototype positions; rather, prototypes are found by exhaustive search optimizing their contribution to the cost function:

$$\boldsymbol{w}_k = \operatorname{argmin}_{\boldsymbol{x}_l} \left\{ \sum_{i,j} \chi_i^*(\boldsymbol{x}_j) \exp(-\mathrm{nd}(i,k)/\sigma^2) d(\boldsymbol{x}_j, \boldsymbol{x}_l) \right\} \tag{7}$$

In the original proposal [19], the summation is restricted to the neighborhood, and possible candidates $\boldsymbol{x}_l$ are restricted to data points mapped to the vicinity of prototype $\boldsymbol{w}_k$. This can be seen as an efficient approximation of the above optimization in particular for small neighborhood range. The choice of (7) has the advantage that convergence of the technique in a finite number of steps can be guaranteed since the algorithm optimizes the cost function of SOM (4) for restricted prototype locations [7].

In complete analogy, batch neural gas can be extended to dissimilarity data by means of the generalized median and the respective cost function. For GTM, a transfer is not possible in general because it is not possible to define a smooth mapping from a continuous latent space to the discrete space of known data points characterized by pairwise dissimilarities.

One important drawback of median approaches is given by the computational complexity: compared to linear time complexity for standard Euclidean topographic mapping, the effort increases to squared complexity due to the necessity of an exhaustive search for every optimization step of the prototypes (7). This can be partially accelerated by means of different techniques such as block summing and branch and bound techniques (see e.g. [14]); due to the dependency of the cost function on all pairwise dissimilarities, every exact technique must inherently be quadratic, however.

Another problematic issue concerns the initialization of median SOM, and its limited capability of smooth updates as compared to standard Euclidean versions. Unlike the Euclidean SOM, an initialization of the map in the direction of the main principal components is hardly possible since only a discrete data space is at our disposable. Due to the rapid convergence of batch techniques, this causes the severe risk that the topographic mapping gets stuck in local optima. Further, as compared to Euclidean settings, less flexibility of the prototypes is available which can cause worse solutions as compared to continuous settings. Tab. 1 shows the results of the techniques for the chromosomes data set, a benchmark from cytogenetics [20]. It consists of 4200 images of chromosomes from 22 classes. Images are compared by aligning strings which describe the thickness of the chromosome profiles in the grey images. Since a labeling is available, an evaluation of the results can be done by posterior labeling of the prototypes according to their receptive field. The test set accuracy (in %) which results from a repeated cross-validation is reported.

median SOM	median NG	AP	relational SOM	relational NG	relational GTM
0.72	0.82	0.9	0.92	0.93	0.92
patch AP	patch RNG (40/10)	Ny RNG (0.01)	patch RGTM (10/5)	Ny RGTM (0.01)	Ny RGTM (0.1)
0.76	0.88	0.93	0.87	0.88	0.55

Table 1. Classification accuracy (in %) on the test set obtained by repeated cross-validation and different clustering techniques on the chromosomes data, the numbers refer to (number of patches / k for k-approximation) for patch processing and (percentage of landmarks) for the Nyström approximation

Obviously, median SOM yields worse results as compared to continuous variants such as relation SOM, which we will explain in the next section. Further, it can be observed that the topological constraint of SOM by the priorly fixed lattice leads to worse results as compared to NG. Interestingly, the accuracy of median techniques is not caused by the restricted representation ability of median clustering, rather numerical problems occur due to the restricted flexibility while optimizing the cost function. This observation is substantiated by the result of affinity propagation (AP) as shown in Tab. 1. AP constitutes an exemplar based clustering scheme which is derived from the quantization error by means of a representation of this cost function as factor graph, and an approximate optimization by means of the max-sum algorithm [10]. Unlike median SOM or median NG, an inherently smooth adaptation process which adapts the likelihood of the data points of becoming an exemplar takes place for AP, resulting in an increased classification accuracy albeit the final solution is represented in terms of data exemplars just as median clustering. AP, however, does not involve any topology such that no topographic mapping is obtained.

4 Relational clustering

As discussed above, the discrete nature of median clustering causes a severe risk to get trapped in local optima of the cost function. Hence the questions arises whether a continuous adaptation of prototypes is possible also for general dissimilarity data. A general approach to extend prototype-based clustering schemes to general dissimilarities has been proposed in [15] in the context of fuzzy clustering, and it has recently been extended in [13] to batch SOM and NG.

Assume that the dissimilarities d_{ij} stem from unknown data in an unknown high dimensional feature vector space, i.e. $d_{ij} = \|\Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{x}_j)\|^2$ for some feature map Φ. Assume that prototypes can be expressed as linear combinations $\boldsymbol{w}_i = \sum_j \alpha_{ij}\Phi(\boldsymbol{x}_j)$ with $\sum_j \alpha_{ij} = 1$. Then, distances can be computed implicitly

$$d(\boldsymbol{w}_i, \boldsymbol{x}_j) = [D\alpha_i]_j - \frac{1}{2} \cdot \alpha_i^t D\alpha_i \tag{8}$$

It has been shown in [13] that this equation also holds if an arbitrary symmetric bilinear form induces dissimilarities in the feature space rather than the squared Euclidean distance.

This observation offers a way to directly transfer SOM and NG to a general symmetric dissimilarity matrix D. As explained e.g. in [13], there always exists a vector space together with a symmetric bilinear form which gives rise to the given dissimilarity matrix. This vector space need not be Euclidean since some eigenvalues associated to the form might be negative or zero. Commonly, this is referred to as pseudo-Euclidean space where the eigenvectors associated to negative eigenvalues serve as a correction to the otherwise Euclidean space. For this vector space, batch NG or SOM can be applied directly in the vector space, and using (8), it can be applied implicitly without knowing the embedding, because of two key issues

1. an implicit representation of prototype $\boldsymbol{w}_i$ in terms of coefficient vectors α_i,
2. Equation (8) to compute the distance in between a data point and a prototype.

Since prototypes as computed by batch NG or SOM can be written as convex combination of data points, and since the update of a prototype depends on the distance only and it decomposes into updates of the coefficients, NG and SOM can be immediately transferred to this setting. So-called relational SOM (RSOM), for example, is given by the iteration of the following steps:

$$\text{compute } d(\boldsymbol{w}_i, \boldsymbol{x}_j) \text{ based on Equation (8)} \tag{9}$$

$$\text{compute } \chi_i^*(\boldsymbol{x}_j) \text{ based on these values} \tag{10}$$

$$\text{adapt } \alpha_{ki} := \frac{\sum_j \chi_i^*(\boldsymbol{x}_j)\cdot\exp(-\mathrm{nd}(i,k))}{\sum_{i,j} \chi_i^*(\boldsymbol{x}_j)\cdot\exp(-\mathrm{nd}(i,k))} \tag{11}$$

Note that this procedure is equivalent to an implicit application of SOM in the pseudo-Euclidean embedding space. It is independent of the concrete embedding and gives the same results if an alternative embedding is used. Further, it is equivalent to standard SOM if a Euclidean embedding of data exists. For general dissimilarities, it constitutes a reasonable extension of SOM to the general case with continuous updates of prototypes.

This procedure, however, has one drawback: albeit it constitutes an exact implementation of SOM in pseudo-Euclidean space, it is no longer clear that the procedure offers an optimization of the corresponding SOM cost function in the embedding space. This is due to the fact that batch SOM itself does not necessarily optimize the cost function in non-Euclidean space; rather, the mean value might constitute a saddle point of the quantization error if data are non-Euclidean. In fact, the quantization error of one receptive field is no longer a convex cost function in the general setting and its optimization is NP hard [25]. Regarding this complexity, the choice (11) can be seen as a reasonable efficient compromise which optimizes the data representation within a receptive field with respect to the positive eigendirections of the underlying bilinear form. See the work [13] for more discussions and experiments concerning this issue. It turns out that the choice (11) hardly deteriorates the value of the cost function in practical applications.

In a similar way, NG can be directly extended to arbitrary dissimilarity data, yielding relational NG (RNG). Similarly, GTM can be extended based on the key observation (8) since also for GTM, prototypes can be chosen as linear combinations of data with coefficients summing up to one. Using Lagrangian functions, it can be proved that this is automatically fulfilled for standard GTM [12]. To realize the approach efficiently, the low dimensional latent space is directly mapped

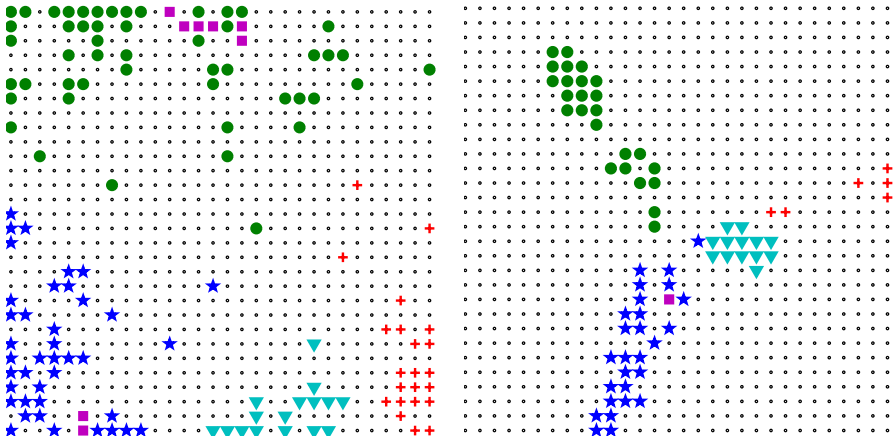

Fig. 1. Visualization of the protein data set incorporating 226 proteins in 5 classes using RSOM (left) and RGTM (right).

to the space of coefficients, see [12]. For relational GTM (RGTM), however, an interpretation by means of a stochastic model is not always clear due to the fact that distances can become negative in pseudo-Euclidean space. For such settings, an interpretation as density values is not obvious; in addition, numerical problems can occur. The publication [12] investigates this setting and demonstrates the feasibility of the approach in several real life examples.

As for the standard Euclidean counterpart, relational GTM tends to display data in a way more suitable for direct data visualization, since less distortions take place for a reasonable number of base functions. One example is shown in Fig. 1. Here protein sequences from different families are compared using an evolutionary distance [22]. In total, 226 globin proteins with 5 different classes are depicted. In both visualizations, the clusters separate according to the a priori known classes. For the RSOM, the prototypes cover the data space with many data being located at the map boundaries, while RGTM widely keeps the internal arrangement due to its stiffness.

In Tab. 1, relational topographic mapping is compared to median approaches, evaluating the techniques in a repeated cross-valiation considering the classification accuracy on the test set for the chromosomes benchmark data. As can be seen from the results, the larger flexibility offered by continuous prototype adaptation in relational topographic mapping leads to an improvement of almost 20% for SOM and almost 20% for NG, arriving at a slightly better value than AP. This fact can be explained by the much simpler numerical optimization of the techniques if a more flexible continuous prototype adaptation is possible instead of only discrete steps. Albeit convergence of relational topographic mapping is not strictly guaranteed (since saddle points might be chosen instead of local optima in case of negative eigenvalues of the corresponding pseudo-Euclidean embedding), divergence never occurred in practical problems.

5 Efficient approximations

Both, median and relational clustering suffer from a quadratic time complexity as compared to linear complexity for their vectorial counterparts. In addition, relational clustering requires linear space complexity since it stores prototypes in terms of coefficient vectors representing the relevance of every data point for the respective prototype. This fact makes the interpretability of the resulting map difficult since it is no longer easily possible to inspect prototypes in the same way as data points. Further, the quadratic time complexity makes the methods infeasible already for medium sized data sets. Different heuristics have recently been proposed in this context to speed up median and relational clustering.

Patch processing constitutes a very simple approach to derive a finite space linear time method based on a prototype based technique. It has been proposed in [1] in the context of the application of NG for streaming data, and, interestingly, it even gives good results if data are not i.i.d. The main idea is to process data consecutively in patches of fixed size. The prototypes counted with multiplicities according to their receptive fields represent all already seen data, and they are included as regular points counted with multiplicities in the next patch. This way, all information is taken into account either directly or in compressed form by means of the prototypes.

If transferred to dissimilarity data, this approach refers to a linear subset of the full dissimilarity matrix only: only those dissimilarities are necessary which correspond to a pair of data in the same patch, further, distances of prototypes representing the previous points and data points in a patch are used. In consequence, an only linear subpart of the full dissimilarity matrix is used this way. Since it is not known a prior which prototypes are used for the topographic mapping, however, the method requires that dissimilarities can be computed instantaneously during the processing. For real life applications this assumption is quite reasonable; e.g. biological sequences can be directly stored and accessed in a data base; their pairwise comparisons can be done on demand using sequence alignment.

Median clustering can directly be extended in a similar way. Unfortunately, such as median topographic mapping itself, it suffers from local optima due to the limited prototype flexibility. In [29], a corresponding extension of affinity propagation is proposed. Due to problems of AP to deal with multiple points, however, the result is worse as compared to AP for the full data set, see Tab. 1.

For relational clustering a direct extension of the patch approach is not possible because prototypes are presented indirectly by referring to the data points. This way, eventually, every prototype refers to all data, i.e. all pairwise dissimilarities have to be known to compute distances in between prototypes and data. In the approach [13], a simple though efficient heuristic is proposed. A prototype is approximated by a fixed number of data points k which are closest to the prototype. These data points are taken to represent the already seen information in compressed form for a new patch. Depending on the value k and the number of patches, a different approximation quality is obtained. Tab. 1 displays the result of relational NG and relational SOM when using patch clustering. As can be seen from the results, a mild degradation of the accuracy (less than 5%) can be observed due to the information loss. The method turns out to be rather robust with respect to the choice of the approximation quality k and the patch size. Further, it can deal with data which are not accessible in an i.i.d. fashion.

The Nyström approximation has been introduced as a standard method to approximate a kernel matrix in [26]. It can be transferred to dissimilarities as presented in [11]. The basic principle is to pick M representative landmarks in the data set which give rise to the rectangular sub-matrix $D_{M,m}$ of dissimilarities of data points and landmarks. This matrix is of linear size, assuming M is fixed. It can be shown (see e.g. [11]) that the full matrix can be approximated in an optimum way in the form

$$D \approx D_{M,m}^t D_{M,M}^{-1} D_{M,m} \tag{12}$$

where $D_{M,M}$ is induced by an $M \times M$ eigenproblem depending on the rectangular submatrices of D. Its computation is $\mathcal{O}(M^3)$ instead of $\mathcal{O}(m^2)$ for the full matrix D. This approximation is exact if M corresponds to the rank of D. It is possible to integrate the approximation (12) in such a way into the distance computation (8) such that the overall effort is linear with respect to m instead of quadratic. This way, a linear approximation technique for relational clustering results. See [11] for detailed formulas. The quality of the result depends very much on the approximation quality of (12), i.e. landmarks should induce a representative dissimilarity matrix. In consequence, the technique is not suited for data which are not i.i.d. For representative landmarks, however, the result can be quite good, as can be seen in Tab. 1: an approximation of the full dissimilarity matrix using only one % of the data as landmarks deteriorates the result not at all for RNG, and by only 4% for RGTM. Interestingly, the result can severely be influenced by the choice of the landmarks: for RGTM, if we pick 10% of tha dat as landmarks, the classification accuracy decreases by nearly 40%. This can be associated to the fact that a highly skewed representation of the dissimilarity matrix is obtained in this case due to the characteristic of the eigenvalue profile of the corresponding dissimilarity matrix. Unlike patch processing, it is fixed a priori which parts of the dissimilarity matrix are relevant for the Nyström method. In consequence, this technique is suited if the dissimilarity matrix D is available a priori, but access to entries of D and the topographic mapping are costly.

As a final demonstration of the feasibility of the approach, we show the result of an experiment in line with the early work of Kohonen for median clustering [19]: GTM is used to visualize a portion of the SWISSPROT data base containing sequences. 10988 sequences according to 32 different functional classes characterized by prosit labels are considered. A GTM with Nyström approximation with 100 landmarks yields the visualization as shown in Fig. 2.

6 Conclusions

We have presented an overview of topographic mapping of dissimilarity data by means of median and relational clustering. Interestingly, popular techniques such as SOM, NG, or GTM can be extended this way, opening the way towards modern data analysis tools for general data formats described in terms of pairwise dissimilarities only. For large data sets, the squared complexity caused by the size of the dissimilarity matrix makes the techniques infeasible already for medium sized data sets. We have presented two techniques to arrive at efficient linear time approximations which offer state of the art linear techniques to deal with large data sets.

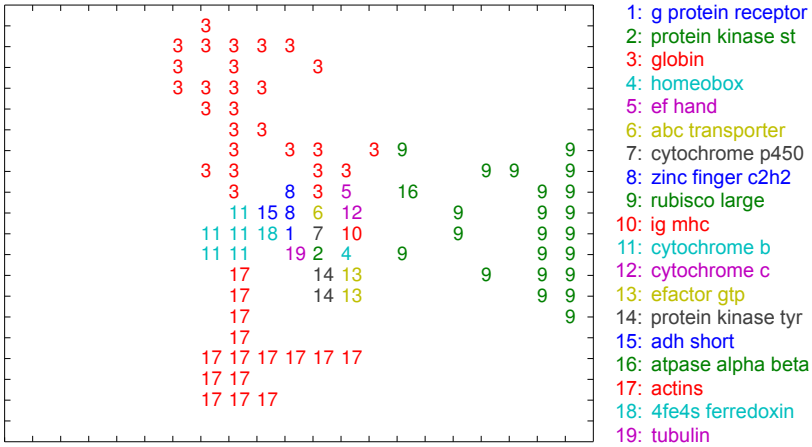

Fig. 2. Around 10000 protein sequences compared by pairwise alignments are depicted on a RGTM trained with the Nyström approximation and 100 landmarks. Posterior labeling displays 19 out of the 32 classes defined by prosit for this data set in a topology preserving manner.

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.

References

1. N. Alex, A. Hasenfuss, and B. Hammer. Patch clustering for massive data sets. *Neurocomputing*, 72(7-9):1455–1469, 2009.
2. C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.
3. C. M. Bishop and C. K. I. Williams. Gtm: The generative topographic mapping. *Neural Computation*, 10:215–234, 1998.
4. 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.
5. L. Bottou and Y. Bengio (1995), Convergence properties of the k-means algorithm, in *NIPS 1994*, 585-592, G. Tesauro, D.S. Touretzky, and T.K. Leen (eds.), MIT.
6. M. Cottrell, J.C. Fort, and G. Pagès (1999), Theoretical aspects of the SOM algorithm, *Neurocomputing* **21:**119-138.
7. M. Cottrell, B. Hammer, A. Hasenfuss, and T. Villmann. Batch and median neural gas. *Neural Networks*, 19:762–771, 2006.
8. R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*, New York: John Wiley & Sons, 2001.
9. J.-C. Fort, P. Letrémy, and M. Cottrell (2002), Advantages and drawbacks of the Batch Kohonen algorithm, in *ESANN'2002*, M. Verleysen (ed.), 223-230, D Facto.
10. B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

11. A. Gisbrecht, B. Mokbel, and B. Hammer. The nystrom approximation for relational generative topographic mappings. In *NIPS workshop on challenges of Data Visualization*, 2010.
12. A. Gisbrecht, B. Mokbel, and B. Hammer. Relational generative topographic map. In M. Verleysen, editor, *ESANN'10*, pages 277–282. D side, 2010.
13. B. Hammer and A. Hasenfuss. Topographic mapping of large dissimilarity datasets. *Neural Computation*, 22(9):2229–2284, 2010.
14. B. Hammer, A. Hasenfuss, and F. Rossi. Median topographic maps for biological data sets. In M. Biehl, B. Hammer, M. Verleysen, and T. Villmann, editors, *Similarity Based Clustering*, Lecture Notes Artificial Intelligence Vol. 5400, pages 92–117. Springer, 2009.
15. R. J. Hathaway and J. C. Bezdek (1994). Nerf c-means: Non-Euclidean relational fuzzy clustering. *Pattern Recognition* 27(3):429-437.
16. T. Heskes (2001). Self-organizing maps, vector quantization, and mixture modeling. *IEEE Transactions on Neural Networks* 12:1299-1305.
17. D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. Visual analytics: Scope and challenges. In S. Simoff, M. H. Boehlen, and A. Mazeika, editors, *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*. Springer, 2008. Lecture Notes in Computer Science (LNCS).
18. T. Kohonen, editor. *Self-Organizing Maps*. Springer-Verlag New York, Inc., 3rd edition, 2001.
19. T. Kohonen and P. Somervuo. How to make large self-organizing maps for non-vectorial data. *Neural Networks*, 15:945–952, 2002.
20. C. Lundsteen, J-Phillip, and E. Granum. Quantitative analysis of 6985 digitized trypsin g-banded human metaphase chromosomes. *Clinical Genetics*, 18:355–370, 1980.
21. T. Martinetz, S. Berkovich, and K. Schulten. Neural-gas Network for Vector Quantization and its Application to Time-Series Prediction. *IEEE-Transactions on Neural Networks*, 4(4):558–569, 1993.
22. H. Mevissen and M. Vingron (1996), Quantifying the local reliability of a sequence alignment, *Protein Engineering* **9:**127-132.
23. M. Neuhaus and H. Bunke (2006), Edit distance based kernel functions for structural pattern classification *Pattern Recognition* 39(10):1852-1863.
24. J.Ontrup and H.Ritter (2001), Hyperbolic slef-organizing maos for semantic navigation, in T.Dietterich, S.Becker, and Z.Ghahramani (eds.), *Advances in Neural Information Processing Systems* 14, pp.1417-1424, MIT Press.
25. P.M. Pardalos and S.A. Vavasis (1991). Quadratic programming with one negative eigenvalue is NP hard. *Journal of Global Optimization* 1:15-22.
26. C. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.
27. H Yin, ViSOM - A novel method for multivariate data projection and structure visualisation,*IEEE Trans. on Neural Networks*, 13(1):237-243, 2002
28. H. Yin. On the equivalence between kernel self-organising maps and self-organising mixture density networks. *Neural Networks*, 19(6-7):780–784, 2006.
29. X. Zhu and B. Hammer. Patch affinity propagation. In *European Symposium on Artificial Neural Networks 2011*, to appear.