Linear time heuristics for topographic mapping of dissimilarity data

Andrej Gisbrecht, Frank-Michael Schleif, Xibin Zhu, and Barbara Hammer

CITEC centre of excellence, Bielefeld University, 33615 Bielefeld - Germany
bhammer@techfak.uni-bielefeld.de

Abstract. Topographic mapping offers an intuitive interface to inspect large quantities of electronic data. Recently, it has been extended to data described by general dissimilarities rather than Euclidean vectors. Unlike its Euclidean counterpart, the technique has quadratic time complexity due to the underlying quadratic dissimilarity matrix. Thus, it is infeasible already for medium sized data sets. We introduce two approximation techniques which speed up the complexity to linear time algorithms: the Nyström approximation and patch processing, respectively. 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. Automated analysis tools offer indispensable techniques to extract relevant information from these data. Popular approaches provide diverse techniques for data structuring and data inspection. Visualization or clustering still constitute one of the most common tasks in this context. Topographic mapping such as offered by the self-organizing map (SOM) [12] and its statistic counterpart, the generative topographic mapping (GTM) [3] provide simultaneous clustering, data visualization, compression by means of prototypes, and inference of the topographic structure of the data manifold in one intuitive framework. For this reason, topographic mapping constitutes a popular tool in diverse areas ranging from remote sensing or biomedical domains up to robotics or telecommunication [12].

Like many classical machine learning techniques, SOM and GTM 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. 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 trees or graph 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.

Quite a few extensions of topographic mapping towards pairwise similarities or dissimilarities have been proposed in the literature. Some are based on a kernelization of existing approaches [4, 18], while others restrict the setting to exemplar based techniques [5, 13]. Some techniques built on alternative cost functions and advanced optimization methods [16, 9]. A very intuitive method which directly extends prototype based clustering to dissimilarity data has been

proposed in the context of fuzzy clustering [11] and later been extended to topographic mapping such as SOM and GTM [10, 8]. Due to its direct correspondence to standard topographic mapping in the Euclidean case, we will focus on the latter techniques. Further, we restrict to the GTM because of its excellent visualization capabilities and its foundation as a stochastic model.

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 two different approximations to speed up GTM for dissimilarities: the Nyström approximation has been proposed in the context of kernel methods as a low rank approximation of the matrix [17]. In [7], preliminary work extends these results to dissimilarities. In this contribution, we demonstrate that the technique provides a suitable linear time approximation for GTM for dissimilarities. As an alternative, patch processing has been proposed in the context of topographic mapping of Euclidean data [1] and later been extended to clustering of dissimilarities [10]. Here we transfer the technique to GTM for dissimilarities, resulting in a linear time method which is even suited if data are not i.i.d. i.e. a representative subpart of the matrix is not accessible priorly.

2 Relational Topographic Mapping

The GTM has been proposed in [3] as a probabilistic counterpart to SOM. It models given data $\mathbf{x} \in \mathbb{R}^D$ 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 $y : \mathbf{w} \mapsto \Phi(\mathbf{w}) \cdot \mathbf{W}$ induced by base functions Φ such as equally spaced Gaussians with bandwidth σ. Every latent point induces a Gaussian

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

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

$$R_{kn}(\mathbf{W}, \beta) = p(\mathbf{w}_k|\mathbf{x}_n, \mathbf{W}, \beta) = \frac{p(\mathbf{x}_n|\mathbf{w}_k, \mathbf{W}, \beta)p(\mathbf{w}_k)}{\sum_{k'} p(\mathbf{x}_n|\mathbf{w}_{k'}, \mathbf{W}, \beta)p(\mathbf{w}_{k'})} \tag{2}$$

of component k for point number n, 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 points $\mathbf{w}_k$, $\mathbf{X}$ to the data points, $\mathbf{R}$ to the responsibilities, and $\mathbf{G}$ is a diagonal matrix with accumulated responsibilities $G_{nn} = \sum_n R_{kn}(\mathbf{W}, \beta)$. The bandwidth is given by

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

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

GTM has been extended to general dissimilarities in [8]. We assume that data $\mathbf{x}$ are given by pairwise dissimilarities $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$ with corresponding dissimilarity matrix D, where the vector representation $\mathbf{x}$ of the data is unknown and $\|\cdot\|^2$ can be induced by any symmetric bilinear form. As pointed out in [11, 10], if prototypes are restricted to linear combinations of the form $\mathbf{t}_k = \sum_{n=1}^{N} \alpha_{kn}\mathbf{x}_n$ with $\sum_{n=1}^{N} \alpha_{kn} = 1$, the prototypes $\mathbf{t}_k$ can be represented indirectly by means of the coefficients $\boldsymbol{\alpha}_k$ and distances can be computed by

$$\|\mathbf{x}_n - \mathbf{t}_k\|^2 = [\mathbf{D}\boldsymbol{\alpha}_k]_n - \frac{1}{2} \cdot \boldsymbol{\alpha}_k^T \mathbf{D}\boldsymbol{\alpha}_k \tag{5}$$

This constitutes the key observation to transfer GTM to relational data $\mathbf{D}$.

As before, targets $\mathbf{t}_k$ induce a Gaussian mixture distribution in the data space. They are obtained as images of points $\mathbf{w}$ in latent space via a generalized linear regression model where, now, the mapping is to the coefficients $y : \mathbf{w}_k \mapsto \boldsymbol{\alpha}_k = \Phi(\mathbf{w}_k) \cdot \mathbf{W}$ with $\mathbf{W} \in \mathbb{R}^{d \times N}$. The restriction $\sum_n [\Phi(\mathbf{w}_k) \cdot \mathbf{W}]_n = 1$ 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 (5). 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 (5) and the optimization of the expectation $\sum_{k,n} R_{kn}(\mathbf{W}_{\text{old}}, \beta_{\text{old}}) \ln p(\mathbf{x}_n | \mathbf{w}_k, \mathbf{W}_{\text{new}}, \beta_{\text{new}})$ 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 $\mathbf{D}$.

3 The Nyström approximation

We shortly review the Nyström technique as presented in [17]. By the Mercer theorem kernels $k(\mathbf{x}, \mathbf{y})$ can be expanded by orthonormal eigenfunctions ϕ_i and non negative eigenvalues λ_i in the form $k(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{y})$. The eigenfunctions and eigenvalues of a kernel are the solution of $\int k(\mathbf{y}, \mathbf{x}) \phi_i(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \lambda_i \phi_i(\mathbf{y})$, which can be approximated based on the Nyström technique by sampling $\mathbf{x}_k$ i.i.d. according to p: $\frac{1}{m} \sum_{k=1}^{m} k(\mathbf{y}, \mathbf{x}_k) \phi_i(\mathbf{x}_k) \approx \lambda_i \phi_i(\mathbf{y})$. Using the matrix eigenproblem $\mathbf{K}^{(m)} \mathbf{U}^{(m)} = \mathbf{U}^{(m)} \mathbf{\Lambda}^{(m)}$ of the $m \times m$ Gram matrix $\mathbf{K}^{(m)}$ we can derive the approximations for the eigenfunctions and eigenvalues

$$\lambda_i \approx \frac{\lambda_i^{(m)}}{m}, \quad \phi_i(\mathbf{y}) \approx \frac{\sqrt{m}}{\lambda_i^{(m)}} \mathbf{k}_y \mathbf{u}_i^{(m)}, \tag{6}$$

where $\mathbf{u}_i^{(m)}$ is the ith column of $\mathbf{U}^{(m)}$. Thus, we can approximate ϕ_i at an arbitrary point $\mathbf{y}$ as long as we know the vector $\mathbf{k}_y = (k(\mathbf{x}_1, \mathbf{y}), ..., k(\mathbf{x}_m, \mathbf{y}))^T$.

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 orthonormal and $\mathbf{\Lambda}$ is diagonal with $\mathbf{\Lambda}_{11} \geq \mathbf{\Lambda}_{22} \geq ... \geq 0$, 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 can approximate a kernel in a similar way, without computing the eigendecomposition of the whole matrix, which is an $O(n^3)$ operation. For a given $n \times n$ Gram matrix $\mathbf{K}$ we randomly choose m rows and respective columns. We denote these rows by $\mathbf{K}_{m,n}$. Using the formulas (6) 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 pseudoinverse,

$$\tilde{\mathbf{K}} = \mathbf{K}_{m,n}^T \mathbf{K}_{m,m}^{-1} \mathbf{K}_{m,n}. \tag{7}$$

This approximation is exact, if $K_{m,m}$ has the same rank as K.

For dissimilarity data, a direct transfer is possible, see [7] for preliminary work on this topic. A symmetric dissimilarity matrix $\mathbf{D}$ is a normal matrix and according to the spectral theorem 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 $d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{y})$ where $\lambda_i \in \mathbb{R}$ correspond to the diagonal elements of $\mathbf{\Lambda}$ and ϕ_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 (7) for the distance matrix, we can apply this result for RGTM. It allows to approximate (5) in the way

$$\|\mathbf{x}_n - \mathbf{t}_k\|^2 \approx \left[\mathbf{D}_{m,n}^T \left(\mathbf{D}_{m,m}^{-1} \left(\mathbf{D}_{m,n}\boldsymbol{\alpha}_k\right)\right)\right]_n - \frac{1}{2} \cdot \left(\boldsymbol{\alpha}_k^T \mathbf{D}_{m,n}^T\right) \cdot \left(\mathbf{D}_{m,m}^{-1} \left(\mathbf{D}_{m,n}\boldsymbol{\alpha}_k\right)\right) \tag{8}$$

with a linear submatrix of m rows and a low rank matrix $\mathbf{D}_{m,m}$ corresponding to the eigenproblem. 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. The last statement holds for constant data space complexity, by means of the eigenproblem and has to be adapted otherwise.

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. A drawback is that a good approximation can only be achieved if the rank of $\mathbf{D}_{m,m}$ is close to the rank of $\mathbf{D}$ as much as possible, i.e. the chosen subset should be representative.

4 Patch Processing

Patch processing takes a different perspective and processes data consecutively in patches of small size m. It has been proposed in [10] in the context of clustering dissimilarity data. Here we present an extension to RGTM.

The principled idea is to compress all already seen data by means of the prototypes as found by RGTM. These prototypes are taken as additional inputs

init:
$E := \emptyset$, number of patch $p := 1$
repeat:
read patch of size m, i.e. $P_{m,m} := \{d_{ij} \mid p \cdot m < i, j \leq (p+1) \cdot m\}$
compute dissimilarities of patch and E,
i.e. $P_{m,|E|} := \{d_{i,j} \mid p \cdot m < i \leq (p+1) \cdot m, \mathbf{x}_j \in E\}$
compute dissimilarities in E, i.e. $P_{|E|,|E|} := \{d_{ij} \mid \mathbf{x}_i, \mathbf{x}_j \in E\}$
this gives the matrix $P := \begin{pmatrix} P_{m,m} & P_{m,|E|} \\ P_{m,|E|}^t & P_{|E|,|E|} \end{pmatrix}$
set the multiplicities to $m_i := \begin{cases} 1 & \text{if } p < i \leq (p+1)m \\ m_i & \text{if } (\mathbf{x}_i, m_i) \in E \end{cases}$
perform RGTM with multiplicities for P
set $E :=$ union of the k nearest data points in P for every prototype $\mathbf{t}_j$
with multiplicities according to their receptive fields

Fig. 1. Principled algorithm for patch RGTM

in the next step in the same way as 'standard' points. Since they compress several data points, they are counted with multiplicities according to the size of their receptive fields. This way, eventually, all data points are processed.

Two extensions are necessary to apply this scheme: we need an efficient realization of RGTM if some data are contained in the training set more than once, i.e. data point $\boldsymbol{x}_i$ comes with multiplicity m_i. Further, since prototypes in RGTM are represented only implicitly by means of coefficient vectors, an efficient approximation of prototype $\mathbf{t}_j$ by means of a priorly fixed number of data points needs to be chosen. Both issues can be dealt with:

- *Extension of RGTM to multiple data points:* Multiple data points affect Eqns. 3, 4. In Eqn. 3, the matrices $\mathbf{G}$ and $\mathbf{R}$ need to weight the responsibilities according to the multiplicities of the data. In Eqn. 4, the summands are weighted by the multiplicities and N is changed accordingly. Similarly, the MDS initialization of RGTM can be extended to multiplicities.
- *Approximation of prototypes by a finite number of points:* fixing the quality k of the approximation, we represent a prototype $\mathbf{t}_j$ by its k closest data points $\mathbf{x}_i$. The union of these data points is taken and every data point is weighted according to the sum of multiplicities of its receptive field.

The algorithm of patch RGTM is displayed in Fig. 1. Since all data are taken into account either directly in the current patch or indirectly represented by the prototypes, processing of data sets in non i.i.d. order is possible. Since it becomes apparent only during training which parts of the dissimilarity matrix are used for training, it is required to compute dissimilarities during training on demand. Only a linear subpart of the dissimilarity corresponding to the size m needs to be considered, hence the algorithm is $\mathcal{O}(m^2 n)$ instead of $\mathcal{O}(n^2)$.

5 Experiments

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

- The *Copenhagen Chromosomes data set* constitutes a benchmark from cytogenetics [14]. A set of 4200 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.
- The *vibrio data set* consists of 1100 samples of vibrio bacteria populations characterized by mass spectra. The spectra encounter approx. 42000 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 [15]. According to the functional form of mass spectra, dedicated similarities as provided by the BioTyper software are used [15].
- Similar to an application presented in [13], we extract roughly 11000 protein sequences of the *SwissProt data base* according to 32 functional labels given by PROSITE [6]. Sequence alignment is done using FASTA [19].

For the chromosomes and vibrio data sets, we use 20×20 prototypes, 10×10 base functions with bandwidth 1, and 50 epochs for training. For patch RGTM, we use 10 patches with a k-approximation with $k \in \{1, 3, 5\}$. For the Nyström approximation, two different fractions of landmarks are tested.

The results of a ten-fold cross-validation with ten repeats are reported in the Tables 1 and 2. The classification accuracy is evaluated using posterior labeling of the prototypes, the standard deviation is given in parenthesis. Further, the CPU time in seconds is reported, the relative speed up as compared to the (not accelerated) RGTM is given in parenthesis. We test the robustness of the techniques with respect to non i.i.d. data by sorting data according to the given class labeling, with only 30 percent random sampling (referred to as 'streaming data'), versus standard random ordering.

Interestingly, the Nyström technique as well as patch processing lead to improved speed (up to a factor 8) on the Chromosome data already for this comparably small data set. The classification accuracy for this data set is only slightly reduced (by less than 5%) for appropriate settings. Obviously, the Nyström technique requires representative sampling while patch processing is more robust against non i.i.d. ordering of data. For the Vibrio data set, no speed up can be achieved using patch processing and the results are massively worse in this case probably due to the fact that a compression of data by few prototypes is not adequately possible. In contrast, the Nyström approximation seems well suited.

For the SwissProt data set we used 40×40 prototypes and bandwidth 0.2. Patch RGTM is done with 11 patches. The results of a ten-fold cross-validation

Chromosome	Classification accuracy	Streaming data	CPU time in sec
RGTM	0.916 (0.003)		2650
RGTM (Nyström 0.01)	0.878 (0.022)	0.626(0.164)	394 (6.7)
RGTM (Nyström 0.1)	0.552 (0.065)	0.365(0.210)	619 (4.3)
Patch RGTM (k=1)	0.845 (0.005)	0.737 (0.023)	318 (8.3)
Patch RGTM (k=3)	0.851 (0.003)	0.777 (0.013)	523 (5.1)
Patch RGTM (k=5)	0.867 (0.004)	0.804 (0.013)	615 (4.3)

Table 1. Results of the methods on the Chromosome data set, standard deviation and speed up are given in parentheses

Vibrio	Classification accuracy	Streaming data	CPU time in sec
RGTM	0.947 (0.005)		78
RGTM (Nyström 0.05)	0.927 (0.005)	0.652 (0.043)	32 (2.4)
RGTM (Nyström 0.1)	0.937 (0.010)	0.590 (0.053)	36 (2.2)
Patch RGTM (k=1)	0.677 (0.020)	0.421 (0.048)	77 (1)
Patch RGTM (k=3)	0.833 (0.012)	0.592 (0.043)	107 (0.7)
Patch RGTM (k=5)	0.889 (0.010)	0.648 (0.044)	149 (0.5)

Table 2. Results on the Vibrio data set reporting standard deviation and speed up in parentheses

with five repeats (only one repeat for RGTM) and the CPU time required to train the map once for the full data set are reported in Table 3. Apparently, the Nyström approximation does not deteriorate the accuracy of the map, while patch processing is not suited due to the incompressibility of the data by few prototypes.

This data set is of medium size, such that the speed up of the Nyström approximation becomes apparent; it accounts for a factor almost 10. Interestingly, also the required memory is widely reduced: in the given example, assuming double precision, about 500 Megabyte are necessary to store the full dissimilarity matrix as compared to about 4.5 Megabyte for the dissimilarities referred to by the Nyström approximation. Using the same number of landmarks and assuming a standard RAM of 12 Gigabyte, this technique would allow to store the required dissimilarities of almost 30 million data points when using the Nyström approximation as compared to only 30 thousand data if the full dissimilarity matrix is required. The speed-up would be in the same order of magnitude due to the dominating factor required to compute the pairwise dissimilarities. Extensions of the technique to a larger fraction of the data set are the subject of ongoing work.

6 Conclusions

Relational GTM offers a highly flexible tool to simultaneously cluster and order dissimilarity data in a topographic mapping. Due to the dependency on the full matrix, the method requires squared time complexity and memory to store the dissimilarities. We have proposed two speed-up techniques which both lead to linear effort: patch processing and the Nyström approximation. Using three examples from the biomedical domain, we demonstrated that already for comparably small data sets the techniques can greatly improve speed while not losing too much information contained in the data.

SwissProt	Classification Accuracy	CPU time in sec
RGTM	0.596	53135
RGTM (Nyström 0.009)	0.630 (0.017)	5892 (9)
Patch RGTM (k=5)	0.388 (0.006)	18623 (2.85)

Table 3. Results on the SwissProt data set; standard deviation, speed up in parentheses

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. 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.
3. C. Bishop, M. Svensen, and C. Williams. The generative topographic mapping. *Neural Computation* 10(1):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. M. Cottrell, B. Hammer, A. Hasenfuss, and T. Villmann. Batch and median neural gas. *Neural Networks*, 19:762–771, 2006.
6. 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.
7. A. Gisbrecht, B. Mokbel, and B. Hammer. The Nystrom approximation for relational generative topographic mappings. In *NIPS workshop on challenges of Data Visualization*, 2010.
8. A. Gisbrecht, B. Mokbel, and B. Hammer. Relational Generative Topographic Mapping. Neurocomputing 74: 1359-1371, 2011.
9. T. Graepel and K. Obermayer (1999), A stochastic self-organizing map for proximity data, *Neural Computation* 11:139-155, 1999.
10. B. Hammer and A. Hasenfuss. Topographic Mapping of Large Dissimilarity Data Sets. Neural Computation 22(9):2229-2284, 2010.
11. R. J. Hathaway and J. C. Bezdek. Nerf c-means: Non-Euclidean relational fuzzy clustering. *Pattern Recognition* 27(3):429-437, 1994.
12. T. Kohonen, editor. *Self-Organizing Maps*. Springer-Verlag New York, Inc., 3rd edition, 2001.
13. T. Kohonen and P. Somervuo (2002), How to make large self-organizing maps for nonvectorial data, *Neural Networks* 15:945-952.
14. C. Lundsteen, J. Phillip, and E. Granum (1980), Quantitative analysis of 6985 digitized trypsin G-banded human metaphase chromosomes, *Clinical Genetics* 18:355-370.
15. T. Maier, S. Klebel, U. Renner, and M. Kostrzewa, Fast and reliable MALDI-TOF ms–based microorganism identification, *Nature Methods*, no. 3, 2006.
16. S. Seo and K. Obermayer (2004), Self-organizing maps and clustering methods for matrix data, *Neural Networks* 17:1211-1230.
17. 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
18. H. Yin. On the equivalence between kernel self-organising maps and self-organising mixture density networks. *Neural Networks*, 19(6-7):780–784, 2006.
19. D. J. Lipman and W. R. Pearson. Rapid and sensitive protein similarity searches. *Science*, 227:1435-1441.