Relational Clustering

Barbara Hammer, Alexander Hasenfuss
Clausthal University of Technology,
Institute of Computer Science

June 7, 2007

Abstract

We introduce relational variants of neural gas, a very efficient and powerful neural clustering algorithm. It is assumed that a similarity or dissimilarity matrix is given which stems from Euclidean distance or dot product, respectively, however, the underlying embedding of points is unknown. In this case, one can equivalently formulate batch optimization in terms of the given similarities or dissimilarities, thus providing a way to transfer batch optimization to relational data. Interestingly, convergence is guaranteed even for general symmetric and nonsingular metrics.

1 Introduction

Topographic maps such as the self-organizing map (SOM) constitute a valuable tool for robust data inspection and data visualization which has been applied in diverse areas such as telecommunication, robotics, bioinformatics, business, etc. [18]. Alternative methods such as neural gas (NG) [22] provide an efficient clustering of data without fixing a prior lattice. This way, subsequent visualization such as multidimensional scaling [21] can readily be applied, whereby no prior restriction of a fixed lattice structure as for SOM is necessary and the risk of topographic errors is minimized. For NG, an optimum (nonregular) data topology is induced such that browsing in a neighborhood becomes directly possible [23].

In the last years, a variety of extensions of these methods has been proposed to deal with more general data structures. This accounts for the fact that more general metrics have to be used for complex data such as microarray data or DNA sequences. Further it might be the case that data are not embedded in a vector space at all, rather, pairwise similarities or dissimilarities are available.

Several extensions of classical SOM and NG to more general data have been proposed: a statistical interpretation of SOM as considered in [5, 14, 30, 31] allows to change the generative model to alternative general data models. The resulting approaches are very flexible but also computationally quite demanding, such that proper initialization and metaheuristics (e.g. deterministic annealing) become necessary when optimizing statistical models. For specific data structures such as time series or recursive structures, recursive models have been proposed as reviewed e.g. in the article [10]. However, these models are restricted

Dagstuhl Seminar Proceedings 07131
Similarity-based Clustering and its Application to Medicine and Biology
http://drops.dagstuhl.de/opus/volltexte/2007/1118

to recursive data structures with Euclidean constituents. Online variants of SOM and NG have been extended to general kernels e.g. in the approaches presented in [27, 34] such that the processing of nonlinearly preprocessed data becomes available. However, these versions have been derived for (slow) online adaptation only.

The approach [20] provides a fairly general method for large scale application of SOM to nonvectorial data: it is assumed that pairwise similarities of data points are available. Then the batch optimization scheme of SOM can be generalized by means of the generalized median to a visualization tool for general similarity data. Thereby, prototype locations are restricted to data points. This method has been extended to NG in [3] together with a general proof of the convergence of median versions of clustering. Further developments concern the efficiency of the computation [2] and the integration of prior information if available to achieve meaningful visualization and clustering [6, 7, 32].

Median clustering has the benefit that it builds directly on the derivation of SOM and NG from a cost function. Thus, the resulting algorithms share the simplicity of batch NG and SOM, its mathematical background and convergence, as well as the flexibility to model additional information by means of an extension of the cost function. However, for median versions, prototype locations are restricted to the set of given training data which constitutes a severe restriction in particular for small data sets. Therefore, extensions which allow a smooth adaptation of prototypes have been proposed e.g. in [8]. In this approach, a weighting scheme is introduced for the points which represents virtual prototype in the space spanned by the training data. This model has the drawback that it is not an extension of the standard Euclidean version.

Here, we use an alternative way to extend NG to relational data given by pairwise Euclidean similarities or dissimilarities, respectively, which is similar to the relational dual of fuzzy clustering as derived in [12, 13]. For a given distance matrix or Gram matrix which stems from a (possibly high-dimensional and unknown) Euclidean space, it is possible to derive the relational dual of topographic map formation which expresses the relevant quantities in terms of the given matrix and which leads to a learning scheme similar to standard batch optimization. This scheme provides identical results as the standard Euclidean version if an embedding of the given data points is known. In particular, it possesses the same convergence properties as the standard variants, thereby restricting the computation to known quantities which do not rely on an explicit embedding. Since these relational variants rely on the same cost function, extensions to additional label information or magnification control [6, 7, 9] become readily available. Further, convergence of the algorithm is guaranteed for every symmetric nonsingular matrix which need not be Euclidean or stem from a metric.

In this contribution, we first introduce batch learning algorithms for neural gas based on a cost function. Then we derive the respective relational dual resulting in a dual cost function and batch optimization schemes for the case of a given distance matrix of data or a given Gram matrix, respectively. We demonstrate the possibility to extend this model to supervised information, and we show the performance in a variety of experiments.

2 Neural gas

Neural clustering and topographic maps constitute effective methods for data preprocessing and visualization. Classical variants deal with vectorial data $\vec{x} \in \mathbb{R}^n$ which are distributed according to an underlying distribution P in the Euclidean plane. The goal of neural clustering algorithms is to distribute prototypes $\vec{w}^i \in \mathbb{R}^n$, $i = 1, \ldots, k$ among the data such that they represent the data as accurately as possible. A new data point $\vec{x}$ is assigned to the *winner* $\vec{w}^{I(\vec{x})}$ which is the prototype with smallest distance $\|\vec{w}^{I(\vec{x})} - \vec{x}\|^2$. This clusters the data space into the receptive fields of the prototypes.

Different popular variants of neural clustering have been proposed to learn prototype locations from given training data [18]. Assume the number of prototypes is fixed to k. Neural gas (NG) [22] optimizes cost function

$$E_{\mathrm{NG}}(\vec{w}) = \frac{1}{2C(\lambda)} \sum_{i=1}^{k} \int h_\lambda(k_i(\vec{x})) \cdot \|\vec{x} - \vec{w}^i\|^2 \, P(d\vec{x})$$

where

$$k_i(\vec{x}) = |\{\vec{w}^j \mid \|\vec{x} - \vec{w}^j\|^2 < \|\vec{x} - \vec{w}^i\|^2\}|$$

is the rank of the prototypes sorted according to the distances, $h_\lambda(t) = \exp(-t/\lambda)$ scales the neighborhood cooperation with neighborhood range $\lambda > 0$, and $C(\lambda)$ is the constant $\sum_{i=1}^{k} h_\lambda(k_i(\vec{x}))$. The neighborhood cooperation smoothes the data adaptation such that, on the one hand, sensitivity to initialization can be prevented, on the other hand, a data optimum topological ordering of prototypes is induced by linking the respective two best matching units for a given data point [23]. Classical NG is optimized in an online mode. For a fixed training set, an alternative fast batch optimization scheme is offered by the following algorithm, which in turn computes ranks, which are treated as hidden variables of the cost function, and optimum prototype locations [3]:

```
init $\vec{w}^i$
repeat
    compute ranks $k_i(\vec{x}^j) = |\{\vec{w}^k \mid \|\vec{x}^j - \vec{w}^k\|^2 < \|\vec{x}^j - \vec{w}^i\|^2\}|$
    compute new prototype locations $\vec{w}^i = \sum_j h_\lambda(k_i(\vec{x}^j)) \cdot \vec{x}^j / \sum_j h_\lambda(k_i(\vec{x}^j))$
```

NG can be used as a preprocessing step for data mining and visualization, followed e.g. by subsequent projection methods such as multidimensional scaling.

It has been shown in e.g. [3] that batch optimization schemes converge in a finite number of steps towards a (local) optimum of the cost function, provided the data points are not located at borders of receptive fields of the final prototype locations. In the latter case, convergence can still be guaranteed but the final solution can lie at the border of basins of attraction.

3 Relational data

Relational data x^i are not explicitely embedded in a Euclidean vector space, rather, pairwise similarities or dissimilarities are available. Batch optimization can be transferred to such situations using the so-called generalized median [3, 20]. Assume, distance information $d(x^i, x^j)$ is available for every pair of data points $x^1, \ldots, x^m$. Median clustering reduces prototype locations to data locations, i.e. adaptation of prototypes is not continuous but takes place within the space $\{x^1, \ldots, x^m\}$ given by the data. We write w^i to indicate that the prototypes need no longer be vectorial. For this restriction, the same cost functions as beforehand can be defined whereby the Euclidean distance $\|\vec{x}^j - \vec{w}^i\|^2$ is substituted by $d(x^j, w^i) = d(x^j, x^{l_i})$ whereby $w^i = x^{l_i}$. Median clustering substitutes the assignment of $\vec{w}^i$ as (weighted) center of gravity of data points by an extensive search, setting w^i to the data points which optimize the respective cost function for fixed assignments. This procedure has been tested e.g. in [3, 6]. It has the drawback that prototypes have only few degrees of freedom if the training set is small. Thus, median clustering usually gives inferior results compared to the classical Euclidean versions when applied in a Euclidean setting.

Here we introduce relational clustering for data characterized by similarities or dissimilarities, using a direct transfer of the standard Euclidean training algorithm to more general settings allowing smooth updates of the solutions. The essential observation consists in a transformation of the cost functions as defined above to their so-called relational dual. We distinguish two settings, similarity data where dot products of training data are available, and dissimilarity data where pairwise distances are available.

3.1 Metric data

Assume training data $x^1, \ldots, x^m$ are given in terms of pairwise distances $d_{ij} = d(x^i, x^j)^2$. We assume that it originates from a Euclidean distance measure, that means, we are always able to find (possibly high dimensional) Euclidean points $\vec{x}^i$ such that $d_{ij} = \|\vec{x}^i - \vec{x}^j\|^2$. Note that this notation includes a possibly nonlinear mapping (feature map) $x^i \mapsto \vec{x}^i$ corresponding to the embedding in a Euclidean space. However, this embedding is not known, such that we cannot directly optimize the above cost functions in the embedding space. The key observation is based on the fact that batch NG optimum prototype locations $\vec{w}^j$ can be expressed as linear combination of data points. Therefore, the unknown values $\|x^j - w^i\|^2$ can be expressed in terms of known values d_{ij}.

More precisely, assume there exist points $\vec{x}^j$ such that $d_{ij} = \|\vec{x}^i - \vec{x}^j\|^2$. Assume the prototypes can be expressed in terms of data points $\vec{w}^i = \sum_j \alpha_{ij} \vec{x}^j$ where $\sum_j \alpha_{ij} = 1$. Then

$$\|\vec{w}^i - \vec{x}^j\|^2 = (D \cdot \alpha_i)_j - 1/2 \cdot \alpha_i^t \cdot D \cdot \alpha_i$$

where $D = (d_{ij})_{ij}$ is the distance matrix and $\alpha_i = (\alpha_{ij})_j$ are the coefficients. This fact can be shown as follows: for $\vec{w}^i = \sum_j \alpha_{ij} \vec{x}^j$, one can compute

$$\|\vec{x}^j - \vec{w}^i\|^2 = \|\vec{x}^j\|^2 - 2 \sum_l \alpha_{il} (\vec{x}^j)^t \vec{x}^l + \sum_{l,l'} \alpha_{il} \alpha_{il'} (\vec{x}^l)^t \vec{x}^{l'} .$$

This is the same as

$$\begin{array}{rl} & (D\cdot\alpha_i)_j - 1/2\cdot\alpha_i^t\cdot D\cdot\alpha_i \\ = & \sum_l \|\vec{x}^j-\vec{x}^l\|^2\cdot\alpha_{il} - 1/2\cdot\sum_{ll'}\alpha_{il}\|\vec{x}^l-\vec{x}^{l'}\|^2\alpha_{il'} \\ = & \sum_l \|\vec{x}^j\|^2\alpha_{il} - 2\cdot\sum_l\alpha_{il}(\vec{x}^j)^t\vec{x}^l + \sum_l\alpha_{il}\|(\vec{x}^l)\|^2 \\ & -\sum_{ll'}\alpha_{il'}\alpha_{il'}\|\vec{x}^l\|^2 + \sum_{ll'}\alpha_{il}\alpha_{il'}(\vec{x}^l)^t\vec{x}^{l'} \end{array}$$

because of $\sum_j \alpha_{ij} = 1$. Because of this fact, we can substitute all terms $\|\vec{x}^j - \vec{w}^i\|^2$ in batch optimization schemes. The parameters α_i yield

1. $\alpha_{ij} = \delta_{i,I(\vec{x}^j)}/\sum_j \delta_{i,I(\vec{x}^j)}$ for k-means,
2. $\alpha_{ij} = h_\lambda(k_i(\vec{x}^j))/\sum_j h_\lambda(k_i(\vec{x}^j))$ for NG

This allows to reformulate the batch optimization in terms of relational data. We obtain

```
init $\alpha_{ij}$ with $\sum_j \alpha_{ij} = 1$
repeat
   compute the distance $\|\vec{x}^j - \vec{w}^i\|^2$ as $(D\cdot\alpha_i)_j - 1/2\cdot\alpha_i^t\cdot D\cdot\alpha_i$
   compute optimum assignments based on this distance matrix
      $\tilde{\alpha}_{ij} = h_\lambda(k_i(\vec{x}^j))$
   compute $\alpha_{ij} = \tilde{\alpha}_{ij}/\sum_j \tilde{\alpha}_{ij}$ as normalization of these values.
```

Hence, prototype locations are computed only indirectly by means of the coefficients α_{ij}. Initialization can be done e.g. setting initial prototype locations to random data points, which is realized by a random selection of k rows from the given distance matrix. Note that prototypes are represented only indirectly by means of the coefficients α_{ij}. For every prototype, m coefficients are stored, m denoting the number of training points. Hence the space complexity of relational clustering is linear w.r.t. the number of training data and the time complexity of one training epoch is quadratic w.r.t. the number of training points.

Given a new data point x which can isometrically be embedded in Euclidean space as $\vec{x}$, and pairwise distances $d_j = d(x,x^j)^2$ corresponding to the distance from x^j, the winner can be determined by using the equality

$$\|\vec{x} - \vec{w}^i\|^2 = (D(x)^t\cdot\alpha_i) - 1/2\cdot\alpha_i^t\cdot D\cdot\alpha_i$$

where $D(x)$ denotes the vector of distances $D(x) = (d_j)_j = (d(x,x^j)^2)_j$.

The quantization error can be expressed in terms of the given values d_{ij} by substituting $\|\vec{x}^j - \vec{w}^i\|^2$ by $(D\cdot\alpha_i)_j - 1/2\cdot\alpha_i^t\cdot D\cdot\alpha_i$. Interestingly, using the formula for optimum assignments of batch optimization, one can also derive relational dual cost functions for the algorithms. We use the abbreviation $k_{ij} = h_\lambda(k_i(\vec{x}^j))$. Because of $\vec{w}^i = \sum_j k_{ij}\cdot\vec{x}^j/\sum_j k_{ij}$, we find

$$\begin{array}{rl} & 1/2\cdot\sum_{ij} k_{ij}\|\vec{x}^j - \vec{w}^i\|^2 \\ = & 1/2\cdot\sum_{ij} k_{ij}\|\vec{x}^j - \sum_l k_{il}\cdot\vec{x}^l/\sum_l k_{il}\|^2 \\ = & \sum_i 1/(2\cdot\sum_l k_{il})\cdot\left(\sum_{ll'} k_{il}k_{il'}\|\vec{x}^l\|^2 - \sum_{ll'} k_{il}k_{il'}(\vec{x}^l)^t\vec{x}^{l'}\right). \end{array}$$

Thus, the relational dual of NG is

$$\sum_i \frac{1}{4\sum_l h_\lambda(k_i(\vec{x}^l))} \cdot \sum_{ll'} h_\lambda(k_i(\vec{x}^l))h_\lambda(k_i(\vec{x}^{l'}))d_{ll'}\,.$$

Note that this relational learning gives exactly the same results as standard batch optimization provided the given relations stem from an Euclidean metric. See e.g. [29] for a characterization of this property. Hence, convergence is guaranteed in this case since it holds for the standard batch versions. If the given distance matrix does not stem from an Euclidean metric, this equality does no longer hold and the terms $(D \cdot \alpha_i)_j - 1/2 \cdot \alpha_i^t \cdot D \cdot \alpha_i$ can become negative. In this case, one can correct the distance matrix by the γ-spread transform $D_\gamma = D + \gamma(\mathbf{1} - \mathbf{I})$ for sufficiently large γ where $\mathbf{1}$ equals 1 for each entry and $\mathbf{I}$ is the identity [12]. For sufficiently large γ, this correction yields a setting where an interpretation of clustering by means of Euclidean prototypes in a possibly high-dimensional Euclidean space exists.

Alternatively, one can apply the formulas for relational clustering directly to any given matrix D, whereby an interpretation by means of explicit prototypes is no longer possible. Interestingly, one can show that this algorithm converges for every symmetric and nonsingular D in a finite number of steps. We present the proof for NG: consider the cost function

$$E(k_{ij},\alpha_{ij}) = \sum_{ij} h_\lambda(k_{ij}) \left(\sum_l d_{jl}\alpha_{il} - \frac{1}{2} \cdot \sum_{ll'} d_{ll'}\alpha_{il}\alpha_{il'} \right)$$

where $\alpha_{ij} \in \mathbb{R}$ and k_{ij} constitutes a permutation of $0, \ldots, k-1$, k denoting the number of prototypes. In relational NG, this cost function is iteratively optimized with respect to α_{ij} for fixed k_{ij} and k_{ij} for fixed α_{ij}. The latter is obvious. The first can be seen as follows: the derivative of $E(k_{ij},\alpha_{ij})$ with respect to α_{nm} yields

$$\sum_j h_\lambda(k_{nj})d_{jm} - \sum_j h_\lambda(k_{nj}) \sum_l d_{lm}\alpha_{nl} = \sum_j d_{jm} \left(h_\lambda(k_{nj}) - \sum_l h_\lambda(k_{nl})\alpha_{nj} \right)$$

For nonsingular D, this is 0 for all n and m iff $\alpha_{nj} = h_\lambda(k_{nj})/\sum_l h_\lambda(k_{nl})$, hence relational NG optimizes α_{nj} in the iterative procedure. Assume $\alpha_{ij}(k_{ij})$ are optimum values α_{ij} for fixed k_{ij}. Assume k_{ij} are given. Assume k'_{ij} are computed in the next iteration of relational NG. Then $E(k_{ij},\alpha_{ij}(k_{ij})) \geq E(k'_{ij},\alpha_{ij}(k_{ij})$ because k'_{ij} is chosen optimum with respect to $\alpha_{ij}(k_{ij}))$, and $E(k'_{ij},\alpha_{ij}(k_{ij})) \geq E(k'_{ij},\alpha_{ij}(k'_{ij}))$ since $\alpha_{ij}(k'_{ij})$ is chosen optimum with respect to k'_{ij}. Hence the cost function decreases in consecutive steps. Since only a finite number of different assignments k_{ij} exists, the algorithm converges (thereby we assume that potential ties for the choice of the ranks k_{ij} are broken deterministically).

Hence relational NG and variants converge for every nonsingular and symmetric matrix D, whereby the cost function $E(k_{ij},\alpha_{ij})$ is minimized. Note that for optimum values $\alpha_{ij} = h_\lambda(k_{ij})/\sum_l h_\lambda(k_{il})$ the cost function yields

$$E(k_{ij},\alpha_{ij}) = \frac{1}{2} \sum_i \frac{1}{\sum_{l''} h_\lambda(k_{il''})} \sum_{ll'} h_\lambda(k_{il})h_\lambda(k_{il'})d_{ll'}\,,$$

i.e. we arrive at the relatioanl dual of NG also when using this procedure for general (symmetric and nonsingular) D.

3.2 Dot products

A dual possibility is to characterize data $x^1, \ldots, x^m$ by means of pairwise similarities, i.e. dot products. We denote the similarity of x^i and x^j by $k(x^i, x^j) = k_{ij}$. We assume that these values fulfill the properties of a dot product, i.e. the matrix K with entries k_{ij} is positive definite. In this case, a representation $\vec{x}^i$ of the data can be found in a possibly high dimensional Euclidean vector space such that $k_{ij} = (\vec{x}^i)^t \vec{x}^j$.

As beforehand, we can represent distances in terms of these values if $\vec{w}^i = \sum_l \alpha_{il} \vec{x}^l$ with $\sum_l \alpha_{il} = 1$ yields optimum prototypes:

$$\|\vec{x}^j - \vec{w}^i\|^2 = k_{jj} - 2\sum_l \alpha_{il} k_{jl} + \sum_{ll'} \alpha_{il}\alpha_{il'} k_{ll'} \, .$$

This allows to compute batch optimization in the same way as beforehand:

```
init $\alpha_{ij}$ with $\sum_j \alpha_{ij} = 1$
repeat
    compute the distance $\|\vec{x}^j - \vec{w}^i\|^2$ as $k_{jj} - 2\sum_l \alpha_{il} k_{jl} + \sum_{ll'} \alpha_{il}\alpha_{il'} k_{ll'}$
    compute optimum assignments based on this distance matrix
        $\tilde{\alpha}_{ij} = h_\lambda(k_i(\vec{x}^j))$ (for NG)
    compute $\alpha_{ij} = \tilde{\alpha}_{ij} / \sum_j \tilde{\alpha}_{ij}$ as normalization of these values.
```

One can use the same identity for $\|\vec{x} - \vec{w}^i\|^2$ to obtain a possibility to compute the winner given a point x and to compute the respective cost function. Convergence of this algorithm is guaranteed since it is identical to the batch versions for the Euclidean data embedding $\vec{x}^i$ if K is positive definite.

If K is not positive definite negative values can occur for $\|\vec{x}^j - \vec{w}^i\|^2$. Then the kernel matrix can be corrected by $K_\gamma = K + \gamma \cdot \mathbf{1}$ with large enough γ.

4 Supervision

The possibility to include further information, if available, is very important to get meaningful results for unsupervised learning. This can help to prevent the 'garbage in - garbage out' problem of unsupervised learning, as discussed e.g. in [16, 17]. Here we assume that additional label information is available which should be accounted for by clustering or visualization. Thereby, labels are embedded in $\mathbb{R}^d$ and can be fuzzy. We assume that the label attached to x^j is denoted by $\vec{y}^j$. We equip a prototype w^i with a label $\vec{Y}^i \in \mathbb{R}^d$ which is adapted during learning. For the Euclidean case, the basic idea consists in a substitution of the standard Euclidean distance $\|\vec{x}^j - \vec{w}^i\|^2$ by a mixture

$$(1-\beta) \cdot \|\vec{x}^j - \vec{w}^i\|^2 + \beta \cdot \|\vec{y}^j - \vec{Y}^i\|^2$$

which takes the similarity of label assignments into account and where $\beta \in [0,1]$ controls the influence of the label values. This procedure has been proposed in

[6, 7, 32] for Euclidean and median clustering and online neural gas, respectively. One can use the same principles to extend relational clustering.

For discrete Euclidean settings $\vec{x}^1, \ldots, \vec{x}^m$ cost functions and related batch optimization is as follows (neglecting constant factors):

$$E_{\mathrm{NG}}(\vec{w}, \vec{Y}) = \sum_{ij} h_\lambda(k_i(\vec{x}^j)) \cdot \Big((1-\beta) \cdot \|\vec{x}^j - \vec{w}^i\|^2 + \beta \cdot \|\vec{y}^j - \vec{Y}^i\|^2\Big)$$

where $k_i(\vec{x}^j)$ denotes the rank of neuron i measured according to the distances $(1-\beta) \cdot \|\vec{x}^j - \vec{w}^i\|^2 + \beta \cdot \|\vec{y}^j - \vec{Y}^i\|^2$. This change in the computation of the rank is accompanied by the adaptation $\vec{Y}^i = \sum_j h_\lambda(\vec{x}^j)\vec{y}^j / \sum_j h_\lambda(\vec{x}^j)$ for the prototype labels for batch optimization

For this generalized cost function, relational learning becomes possible by substituting the distances $\|\vec{x}^j - \vec{w}^i\|^2$ using the identity $\vec{w}^i = \sum \alpha_{ij}\vec{x}^j$ for optimum assignments which still holds for these extensions. The same computation as beforehand yields to the algorithm for clustering dissimilarity data characterized by pairwise distances d_{ij}:

```
init $\alpha_{ij}$ with $\sum_j \alpha_{ij} = 1$
repeat
    compute the distances as $(1-\beta) \cdot ((D \cdot \alpha_i)_j - 1/2 \cdot \alpha_i^t D \alpha_i) + \beta \cdot \|Y^i - y^j\|^2$
    compute optimum assignments $\tilde{\alpha}_{ij}$ based on this distance as before
    compute $\alpha_{ij} = \tilde{\alpha}_{ij} / \sum_j \tilde{\alpha}_{ij}$
    compute prototype labels $\vec{Y}^i = \sum_j \alpha_{ij}\vec{y}^j$
```

An extension to similarity data given by dot products $\vec{x}^i \cdot \vec{x}^j$ proceeds in the same way using the distance computation based on dot products as derived beforehand. As beforehand, this version converges in a finite number of steps.

5 Experiments

In the experiments, we focus on the clustering and classification ability of the algorithms rather than the visualization, since these aspects can easily be evaluated by the classification error for given data labels. For comparison, we also include k-means which is obrained in the limit of vanishing neighborhood cooperation. We demonstrate the performance of the neural gas and k-means algorithms in different scenarios covering a variety of characteristic situations. All algorithms have been implemented based on the SOM Toolbox for Matlab [26]. Note that, for all median versions, prototypes situated at identical points of the data space do not separate in subsequent runs. Therefore constellations with exactly identical prototypes should be avoided. For the Euclidean and relational versions this problem is negligible, presumed prototypes are initialized at different positions. However, for median versions it is likely that prototypes move to an identical locations due to the limited number of different positions in data space, in particular for small data sets. To cope with this fact in median versions, we add a small amount of noise to the distances in each epoch in

	k-Means	Supervised k-Means	Median k-Means	Relational k-Means	Supervised Relational k-Means
Accuracy					
Mean	93.6	93.0	93.0	93.4	93.5
StdDev	0.8	1.1	1.0	1.2	1.1
	Batch NG	Supervised Batch NG	Median Batch NG	Relational Batch NG	Supervised Relational Batch NG
Accuracy					
Mean	94.1	94.7	93.1	94.0	94.4
StdDev	1.0	0.8	1.0	0.9	1.0

Table 1: Classification accuracy on the WDBC database for posterior labeling. The mean accuracy over 100 repeats of 2-fold cross-validation is reported.

order to separate identical prototypes. The initial neighborhood rate for neural gas is $\lambda = n/2$, n being the number of neurons, and it is multiplicatively decreased during training. In all runs, relational clustering has been applied directly without any correction of the given matrix.

Wisconsin Breast Cancer Database

The Wisconsin Diagnostic Breast Cancer database (WDBC) is a standard benchmark set from clinical proteomics [33]. It consists of 569 data points described by 30 real-valued input features: digitized images of a fine needle aspirate of breast mass are described by characteristics such as form and texture of the cell nuclei present in the image. Data are labeled by two classes, benign and malignant.

For training we used 40 neurons and 150 epochs per run. The dataset was z-transformed beforehand. The results were gained from repeated 2-fold cross-validations averaged over 100 runs. The mixing parameter of the supervised methods was set to 0.5 for the simulations reported in Table 1. Moreover, the data set is contained in the Euclidean space therefore we are able to compare the relational versions introduced in this article to the standard Euclidean methods. These results are shown in Table 1. The effect of a variation of the mixing parameter is demonstrated in Fig. 1. The results are competitive to supervised learning with the state-of-the-art-method GRLVQ as obtained in [28].

As one can see, the results of Euclidean and relational clustering are identical, as expected by the theoretical background of relational clustering. Relational clustering and supervision allow to improve the more restricted and unsupervised median versions by more than 1% classification accuracy.

Cat Cortex

The Cat Cortex Data Set originates from anatomic studies of cats' brains. A matrix of connection strengths between 65 cortical areas of cats was compiled from literature [4]. There are four classes corresponding to four different regions

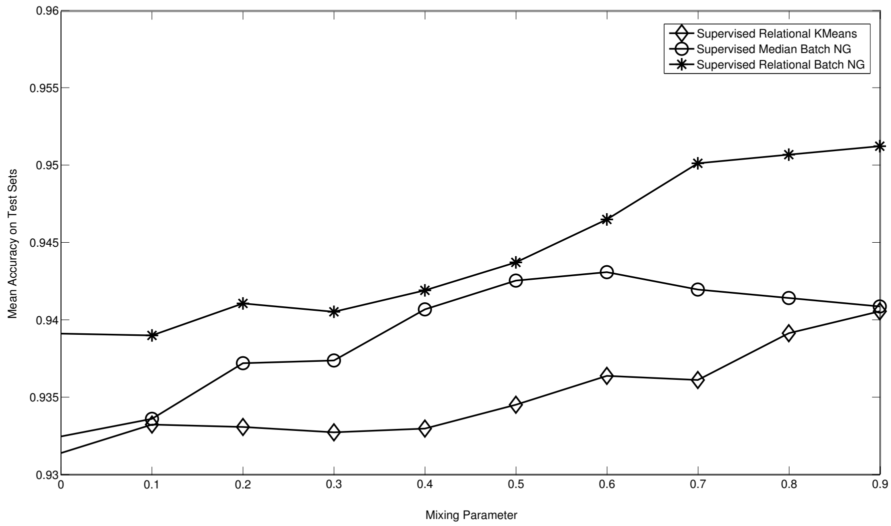

Figure 1: Results of the supervised methods for the WDBC data set with different mixing parameters applied.

of the cortex. For our experiments a preprocessed version of the data set from Haasdonk et al. [11] was used. The matrix is symmetric but the triangle inequality does not hold.

The algorithms were tested in 10-fold cross-validation using 12 neurons (three per class) and 150 epochs per run. The results presented reveal the mean accuracy over 250 repeated 10-fold cross-validations per method. The mixing parameter of the supervised methods was set to 0.5 for the simulations reported in Table 2. Results for different mixing parameters are shown in Figure 2.

A direct comparison of our results to the findings of Graepel et al. [4] or Haasdonk et al. [11] is not possible. Haasdonk et al. gained an accumulated error over all classes of at least 10% in leave-one-out experiments with SVMs. Graepel et al. obtained virtually the same results with the Optimal Hyperplane (OHC) algorithm. In our experiments, the improvement of restricted median

	Median k-Means	Median Batch NG	Relational k-Means	Relational Batch NG	Supervised Median Batch NG	Supervised Relational k-Means	Supervised Relational Batch NG
Accuracy							
Mean	72.8	71.6	89.0	88.7	77.9	89.2	91.3
StdDev	3.9	4.0	3.3	3.0	3.5	3.0	2.8

Table 2: Classification accuracy on the Cat Cortex Data Set for posterior labeling. The mean accuracy over 250 repeats of 10-fold cross-validation is reported.

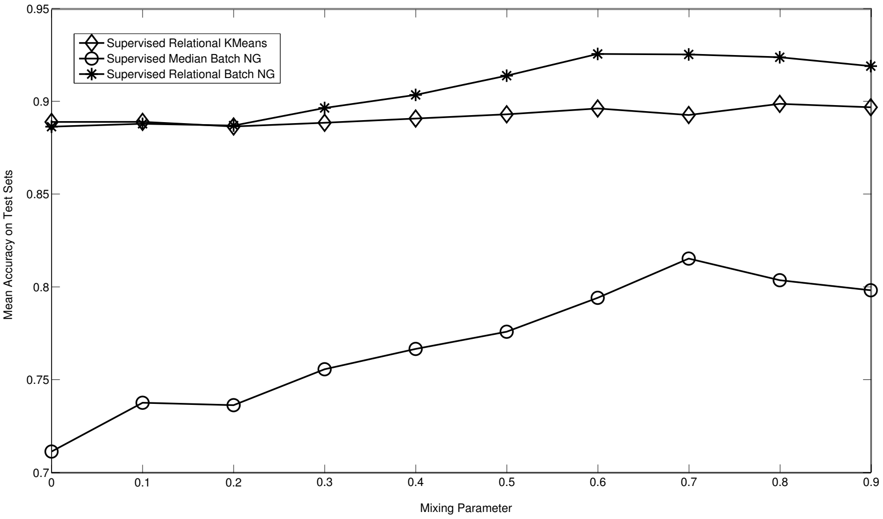

Figure 2: Results of the supervised methods for the Cat Cortex Data Set with different mixing parameters applied.

clustering by relational extensions can clearly be observed, which accounts for more than 10% classification accuracy. Note that relational clustering works quite well in this case although a theoretical foundation due to the non-metric similarity matrix is missing.

Proteins

The evolutionary distance of 226 globin proteins is determined by alignment as described in [24]. These samples originate from different protein families: hemoglobin-α, hemoglobin-β, myoglobin, etc. Here, we distinguish five classes as proposed in [11]: HA, HB, MY, GG/GP, and others.

For training we used 45 neurons and 150 epochs per run. The results were gained from repeated 10-fold cross-validations averaged over 100 runs. The mixing parameter of the supervised methods was set to 0.5 for the simulations reported in Table 3.

Unlike the results reported in [11] for SVM which uses one-versus-rest encoding, the classification in our setting is given by only one clustering model. Depending on the choice of the kernel, [11] reports errors which approximately add up to 4% for the leave-one-out error. This result, however, is not comparable to our results due to the different error measure. A 1-nearest neighbor classifier yields an accuracy 91.6 for our setting (k-nearest neighbor for larger k is worse; [11] which is comparable to our results.

Chromosomes

The Copenhagen chromosomes database is a benchmark from cytogenetics [19]. A set of 4200 human nuclear chromosomes from 22 classes (the X resp. Y sex chromosome is not considered) are represented by the grey levels of their images and transferred to strings representing the profile of the chromosome by the thickness of their silhouettes. Thus, this data set consists of strings of different length, and standard k-means clustering cannot be used. Median versions, however, are directly applicable. The edit distance is a typical distance measure for two strings of different length, as described in [15, 25]. In our application, distances of two strings are computed using the standard edit distance whereby substitution costs are given by the signed difference of the entries and insertion/deletion costs are given by 4.5 [25].

The algorithms were tested in 2-fold cross-validation using 100 neurons and 100 epochs per run (cf. [3]). The results presented are the mean accuracy over 10 times 2-fold cross-validation per method. The mixing parameter of the supervised methods was set to 0.9.

As can be seen, supervised relational neural gas achieves an accuracy of 0.914 for $\alpha = 0.9$. This improves by 8% compared to median variants.

6 Discussion

We have introduced relational neural clustering which extends the classical Euclidean versions to settings where pairwise distances or dot products of the data are given but no explicit embedding into a Euclidean space is known. By means of the relational dual, batch optimization can be formulated in terms of these quantities only. This extends previous median clustering variants to a continuous prototype update which is particularly useful for only sparsely sampled data. The derived relational algorithms have a formal background only for Euclidean distances or metrics; however, as demonstrated in an example for the cat cortex data, the algorithms might also prove useful in more general scenarios, and converegence is guaranteed for fairly general settings. In all expreminats presented inthis contribution, relational clustering significantly improves the classification accuracy obtained by semi-supervised clustering compared to median clustering using the same underlying cost function. Depending on the data set at hand, results which are competitive to state-of-the-art classification (using dedicated supervised training) could be approximated in our settigns,

	Median k-Means	Median Batch NG	Relational k-Means	Relational Batch NG	Supervised Median Batch NG	Supervised Relational k-Means	Supervised Relational Batch NG
Accuracy							
Mean	76.1	76.3	88.0	89.9	89.4	88.2	90.0
StdDev	1.3	1.8	1.8	1.3	1.4	1.7	1.0

Table 3: Classification accuracy on the Protein Data Set for posterior labeling. The mean accuracy over 100 repeats of 10-fold cross-validation is reported.

	Median k-Means	Median Batch NG	Relational k-Means	Relational Batch NG	Supervised Median Batch NG	Supervised Relational k-Means	Supervised Relational Batch NG
Accuracy							
Mean	82.3	82.8	90.6	91.3	89.4	90.1	91.4
StdDev	2.2	1.7	0.6	0.2	0.6	0.6	0.6

Table 4: Classification accuracy on the Copenhagen Chromosome Database for posterior labeling. The mean accuracy over 10 runs of 2-fold cross-validation is reported.

demonstrating the efficiency and robustness of relational clustering. However, being based on the quantization error and related quantities, relational clustering is mainly intended for data inspection whereby additional information can be integrated to achieve meaningful clusters. The general framework as introduced in this article opens the way towards the transfer of further principles of SOM and NG to the setting of relational data: as an example, the magnification factor of topographic map formation for relational data transfers from the Euclidean space, and possibilities to control this factor as demonstrated for batch clustering e.g. in the approach [9] can readily be used.

One very important subject of future work concerns the complexity of computation and sparseness of prototype representation. For the approach as introduced above, the complexity scales quadratic with the number of training examples and the size of prototype representations is linear with respect to the number of examples. The representation contains a large number of very small coefficients, which correspond to data points for which the distance from the prototype is large. Therefore it can be expected that a restriction of the representation to the close neighborhood is sufficient for accurate results.

References

[1] James W. Anderson (2005), Hyperbolic Geometry, second edition, Springer.

[2] B. Conan-Guez, F. Rossi, and A. El Golli (2005), A fast algorithm for the self-organizing map on dissimilarity data, in *Workshop on Self-Organizing Maps*, 561-568.

[3] M. Cottrell, B. Hammer, A. Hasenfuss, and T. Villmann (2006), Batch and median neural gas, *Neural Networks*, 19:762-771.

[4] T. Graepel, R. Herbrich, P. Bollmann-Sdorra and K. Obermayer (1999), Classification on pairwise proximity data, In M. I. Jordan, M. J. Kearns, and S. A. Solla (Eds.), NIPS, vol. 11, MIT Press, p. 438-444.

[5] T. Graepel and K. Obermayer (1999), A stochastic self-organizing map for proximity data, *Neural Computation* 11:139-155.

[6] B. Hammer, A. Hasenfuss, F.-M. Schleif, and T. Villmann (2006), Supervised median neural gas, In Dagli, C., Buczak, A., Enke, D., Embrechts, A., and Ersoy, O. (Eds.), *Intelligent Engineering Systems Through Artificial Neural Networks* 16, Smart Engineering System Design, pp.623-633, ASME Press.

[7] B. Hammer, A. Hasenfuss, F.-M. Schleif, and T. Villmann (2006), Supervised batch neural gas, In *Proceedings of Conference Artificial Neural Networks in Pattern Recognition (ANNPR)*, F. Schwenker (ed.), Springer, pages 33-45.

[8] A. Hasenfuss, B. Hammer, F.-M. Schleif, and T. Villmann (2007), Neural gas clustering for dissimilarity data with continuous prototypes, accepted for IWANN'07.

[9] B. Hammer, A. Hasenfuss, and T. Villmann (2007), Magnification control for batch neural gas, *Neurocomputing* 70:1225-1234.

[10] B. Hammer, A. Micheli, A. Sperduti, and M. Strickert (2004), Recursive self-organizing network models, *Neural Networks* 17(8-9):1061-1086.

[11] B. Haasdonk and C. Bahlmann (2004), Learning with distance substitution kernels, in *Pattern Recognition - Proc. of the 26th DAGM Symposium.*

[12] R. J. Hathaway and J. C. Bezdek. Nerf c-means: Non-euclidean relational fuzzy clustering. *Pattern Recognition* 27(3):429-437, 1994.

[13] R. J. Hathaway, J. W. Davenport, and J. C. Bezdek. Relational duals of the c-means algorithms. *Pattern Recognition* 22:205-212, 1989.

[14] T. Heskes (2001). Self-organizing maps, vector quantization, and mixture modeling. *IEEE Transactions on Neural Networks* 12:1299-1305

[15] A. Juan and E. Vidal (2000), On the use of normalized edit distances and an efficient k-NN search technique (k-AESA) for fast and accurate string classification, in *ICPR 2000*, vol.2, p. 680-683.

[16] S. Kaski, J. Nikkilä, E. Savia, and C. Roos (2005), Discriminative clustering of yeast stress response, In *Bioinformatics using Computational Intelligence Paradigms*, U. Seiffert, L. Jain, and P. Schweizer (eds.), pages 75-92, Springer.

[17] S. Kaski, J. Nikkilä, M. Oja, J. Venna, P. Törönen, and E. Castren (2003), Trustworthiness and metrics in visualizing similarity of gene expression, *BMC Bioinformatics*, 4:48.

[18] T. Kohonen (1995), *Self-Organizing Maps*, Springer.

[19] C. Lundsteen, J. Phillip, and E. Granum (1980), Quantitative analysis of 6985 digitized trypsin G-banded human metaphase chromosomes, *Clinical Genetics* **18**:355-370.

[20] T. Kohonen and P. Somervuo (2002), How to make large self-organizing maps for nonvectorial data, *Neural Networks* 15:945-952.

[21] J. B. Kruskal (1964), Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29:1-27.

[22] T. Martinetz, S.G. Berkovich, and K.J. Schulten (1993), 'Neural-gas' network for vector quantization and its application to time-series prediction, *IEEE Transactions on Neural Networks* 4:558-569.

[23] T. Martinetz and K. Schulten (1994), Topology representing networks. *Neural Networks* 7:507-522.

[24] H. Mevissen and M. Vingron (1996), Quantifying the local reliability of a sequence alignment, *Protein Engineering* **9:**127-132.

[25] M. Neuhaus and H. Bunke (2006), Edit distance based kernel functions for structural pattern classification *Pattern Recognition* 39(10):1852-1863.

[26] Neural Networks Research Centre, Helsinki University of Technology, SOM Toolbox, http://www.cis.hut.fi/projects/somtoolbox/

[27] A.K. Qin and P.N. Suganthan (2004), Kernel neural gas algorithms with application to cluster analysis, *ICPR 2004* vol.4, pp.617-620.

[28] F.-M. Schleif, B. Hammer, and T. Villmann (2007), Margin based Active Learning for LVQ Networks, *Neurocomputing* 70(7-9):1215-1224.

[29] B. Schölkopf (2000), The kernel trick for distances, Microsoft TR 2000-51.

[30] S. Seo and K. Obermayer (2004), Self-organizing maps and clustering methods for matrix data, *Neural Networks* 17:1211-1230.

[31] P. Tino, A. Kaban, and Y. Sun (2004), A generative probabilistic approach to visualizing sets of symbolic sequences. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD-2004*, (eds) R. Kohavi, J. Gehrke, W. DuMouchel, J. Ghosh. pp. 701-706, ACM Press.

[32] T. Villmann, B. Hammer, F. Schleif, T. Geweniger, and W. Herrmann (2006), Fuzzy classification by fuzzy labeled neural gas, *Neural Networks*, 19:772-779.

[33] W.H. Wolberg, W.N. Street, D.M. Heisey, and O.L. Mangasarian (1995), Computer-derived nuclear features distinguish malignant from benign breast cytology, *Human Pathology*, **26**:792–796.

[34] H. Yin (2006), On the equivalence between kernel self-organising maps and self-organising mixture density network, *Neural Networks* 19(6):780-784.