Patch Relational Neural Gas – Clustering of Huge Dissimilarity Datasets

Alexander Hasenfuss[1], Barbara Hammer[1], and Fabrice Rossi[2]

[1] Clausthal University of Technology, Department of Informatics,
Clausthal-Zellerfeld, Germany

[2] Projet AxIS, INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt,
B.P. 105, 78153 Le Chesnay Cedex, France

Abstract. Clustering constitutes an ubiquitous problem when dealing with huge data sets for data compression, visualization, or preprocessing. Prototype-based neural methods such as neural gas or the self-organizing map offer an intuitive and fast variant which represents data by means of typical representatives, thereby running in linear time. Recently, an extension of these methods towards relational clustering has been proposed which can handle general non-vectorial data characterized by dissimilarities only, such as alignment or general kernels. This extension, relational neural gas, is directly applicable in important domains such as bioinformatics or text clustering. However, it is quadratic in m both in memory and in time (m being the number of data points). Hence, it is infeasible for huge data sets. In this contribution we introduce an approximate patch version of relational neural gas which relies on the same cost function but it dramatically reduces time and memory requirements. It offers a single pass clustering algorithm for huge data sets, running in constant space and linear time only.

1 Introduction

The presence of huge data sets, often several GB or even TB, poses particular challenges towards standard data clustering and visualization such as neural gas or the self-organizing map [10, 12]. At most a single pass over the data is still affordable such that online adaptation which requires several runs over the data is not applicable. At the same time, alternative fast batch optimization cannot be applied due to memory constraints. In recent years, researchers have worked on so-called single pass clustering algorithms which run in a single or few passes over the data and which require only a priorly fixed amount of allocated memory. Popular methods include heuristics such as CURE, STING, and BIRCH [5, 16, 18] and approximations of k-means clustering as proposed in [4, 9]. In addition, dynamic methods such as growing neural gas have been adapted to cope with the scenario of life-long adaptivity, see e.g. [15].

The situation becomes even more complicated if data are non-vectorial and distance-based clustering methods have to be applied, which often display a quadratic time complexity [8]. Although a variety of methods which can directly work with relational data based on general principles such as extensions of the self-organizing map and neural gas have been proposed [11, 3, 7], these methods are not suited for huge data sets. For complex metrics such as alignment of DNA strings or complex kernels for text data, it is infeasable to compute all pairs of the distance matrix and at most a small fraction can effectively be addressed. A common challenge today, arising especially in Computational Biology, are huge datasets whose pairwise dissimilarities cannot be hold at once within random-access memory during computation, due to the sheer amount of data.

In this work, we present a new technique based on the Relational Neural Gas approach [7] that is able to handle this situation by a single pass technique based on patches that can be chosen in accordance to the size of the available random-access memory. This results in a linear time and finite memory algorithm for general dissimilarity data which shares the intuitivity and robustness of NG.

2 Neural Gas

Neural Gas (NG), introduced by Martinetz et al. [12], is a vector quantization technique aiming for representing given data $v \in V \subseteq \mathbb{R}^d$ faithfully by prototypes $w_i \in \mathbb{R}^d$, $i = 1, \ldots, n$. For a continuous input distribution given by a probability density function $P(v)$, the cost function minimized by NG is

$$E \sim \frac{1}{2} \sum_{i=1}^{n} \int h_\lambda(k(w_i, v)) \cdot \|v - w_i\|^2 \, P(v) dv,$$

where $k(w_i, v) = |\{w_j : \|v - w_j\| < \|v - w_i\|\}|$ denotes the rank of neuron w_i arranged according to the distance from data point v. The parameter $\lambda > 0$ controls the neighbourhood range through the exponential function $h_\lambda(t) = \exp(-t/\lambda)$.

Typically, NG is optimized in an online mode using a stochastic gradient descent method. However, for a given discrete training set $\{v_1, v_2, \ldots, v_m\}$ the cost function of NG becomes

$$E(W) \sim \frac{1}{2} \cdot \sum_{i=1}^{n} \sum_{j=1}^{m} h_\lambda(k(w_i, v)) \cdot \|v_j - w_i\|^2 \tag{1}$$

For this case, an alternative batch optimization technique has been introduced [3]. It, in turn, determines the ranks $k_{ij} = k(w_i, v_j)$ for fixed prototype locations w_i and then determines new prototype locations via the update formula

$$w_i = \sum_j h_\lambda(k_{ij}) \cdot v_j / \sum_j h_\lambda(k_{ij})$$

for the fixed ranks k_{ij}. Batch NG shows the same accuracy and behaviour as NG, whereby its convergence is quadratic instead of linear as for NG.

3 Relational Neural Gas

Relational data do not necessarily originate from an Euclidean vector space, instead only a pairwise dissimilarity measure d_{ij} is given for the underlying datapoints $v_i, v_j \in V$. The only demands made on dissimilarity measures are non-negativity $d_{ij} \geq 0$ and reflexivity $d_{ii} = 0$, so they are not necessarily metric or even symmetric by nature. Obviously, NG cannot directly deal with such data and its original formulation is restricted to vectorial updates.

One way to deal with relational data is Median clustering [3]. This technique restricts prototype locations to given data points, such that distances are well defined in the cost function of NG. Batch optimization can be directly tranferred to this case. However, median clustering has the inherent drawback that only discrete adaptation steps can be performed which can dramatically reduce the representation quality of the clustering.

Relational Neural Gas (RNG) [7] overcomes the problem of discrete adaptation steps by using convex combinations of Euclidean embedded data points as prototypes. For that purpose, we assume that there exists a set of (in general unknown and presumably high dimensional) Euclidean points V such that $d_{ij} = \|v_i - v_j\|$ for all $v_i, v_j \in V$ holds, i.e. we assume there exists an (unknown) isometric embedding into an Euclidean space. The key observation is based on the fact that, under the assumptions made, the squared distances $\|w_i - v_j\|^2$ between (unknown) embedded data points and optimum prototypes can be expressed merely in terms of known distances d_{ij}.

In detail, we express the prototypes as $w_i = \sum_j \alpha_{ij} v_j$ with $\sum_j \alpha_{ij} = 1$. With optimal prototypes, this assumption is necessarily fulfilled. Given a coefficient matrix $(\alpha_{ij}) \in \mathbb{R}^{n \times m}$ and a matrix $\Delta = \left(d_{ij}^2\right) \in \mathbb{R}^{m \times m}$ of squared distances, it then holds

$$\|w_i - v_j\|^2 = (\alpha_{i*} \cdot \Delta)_j - \frac{1}{2} \cdot \alpha_{i*} \Delta \alpha_{i*}^T \tag{2}$$

where $*$ indicates vector indices. Because of this fact, we are able to substitute all terms $\|w_i - v_j\|^2$ in Batch NG by (2) and derive new update rules. For optimum prototype locations given fixed ranks we find

$$\alpha_{ij} = h_\lambda(k_i(v_j)) / \sum_t h_\lambda(k_i(v_t)). \tag{3}$$

This allows to reformulate the batch optimization schemes in terms of relational data as done in [7].

Note that, if an isometric embedding into Euclidean space exists, this scheme is equivalent to Batch NG and it yields identical results. Otherwise, the consecutive optimization scheme can still be applied. It has been shown in [7] that Relational NG converges for every nonsingular symmetric matrix Δ and it optimizes the relational dual cost function of NG which can be defined solely based on distances Δ.

Relational neural gas displays very robust results in several applications as shown in [7]. Compared to original NG, however, it has the severe drawback that the computation time is $\mathcal{O}(m^2)$, m being the number of data points, and the required space is also quadratic (because of Δ). Thus, this method becomes infeasible for huge data sets. Recently, an intuitive and powerful method has been proposed to extend batch neural gas towards a single pass optimization scheme which can be applied even if the training points do not fit into the main memory [1]. The key idea is to process data in patches, whereby prototypes serve as a sufficient statistics of the already processed data. Here we transfer this idea to relational clustering.

4 Patch Relational Neural Gas

Assume as before that data are given as a dissimilarity matrix $D = (d_{ij})_{i,j=1,\ldots,m}$ with entries $d_{ij} = d(v_i, v_j)$ representing the dissimilarity of the datapoints v_i and v_j. During processing of Patch Relational NG, n_p patches of fixed size $p = \lfloor m/n_p \rfloor$ are cutted consecutively from the dissimilarity matrix D[3], where every patch

$$P_i = (d_{st})_{s,t=(i-1)\cdot p+1,\ldots,i\cdot p} \in \mathbb{R}^{p\times p}$$

is a submatrix of D centered around the matrix diagonal.

The idea of the original patch scheme is to add the prototypes from the processing of the former patch P_{i-1} as additional datapoints to the current patch P_i, forming an extended patch P_i^* which includes the previous points in the form of a compressed statistics. The additional datapoints – the former prototypes – are weighted according to the size of their receptive fields, i.e. how many datapoints do they represent in the former patch. To implement this fact, every datapoint v_j is equipped with a multiplicity m_j, which is initialized with $m_j = 1$ for data points from the training set and it is set to the size of the receptive fields for data points stemming from prototypes. This way, all data are processed without loss of previous information which is represented by the sufficient statistics. So far, the method has only been tested for stationary distributions. However, it can expected that the method works equally well for nonstationary distributions due to the weighting of already processed information according to the number of already seen data points. In contrast to dynamic approaches such as [15] the number of prototypes can be fixed a priori.

Unlike the situation of original Patch NG [1], where prototypes can simply be converted to datapoints and the inter-patch distances can always be recalculated using the Euclidean metric, the situation becomes more difficult for relational clustering. In Relational NG prototypes are expressed as convex combinations of unknown Euclidean datapoints, only the distances can be calculated. Moreover, the relational prototypes gained from processing of a patch cannot be simply converted to datapoints for the next patch. They are defined only on the datapoints of the former patch. To calculate the necessary distances between these prototypes and the datapoints of the next patch, the distances between former and next patch must be taken into account, as shown in [7]. But that means touching all elements of the upper half of the distance matrix at least once during processing of all patches, what foils the idea of the patch scheme to reduce computation and memory-access costs.

In this contribution, another way is proposed. In between patches not the relational prototypes itselves but representative datapoints obtained from a so

[3] The remainder is no further considered here for simplicity. In the practical implementation the remaining datapoints are simply distributed over the first $(M - p \cdot n_p)$ patches.

called k-approximation are used to extend the next patch. As for standard patch clustering, the points are equipped with multiplicities. On each extended patch a modified Relational NG is applied taking into account the multiplicities.

***k*-Approximation** Assume there are given n relational prototypes by their coefficient matrix $(\alpha_{ij}) \in \mathbb{R}^{n \times m}$ defined on Euclidean datapoints V. These prototypes are taken after convergence of the Relational NG method, i.e. these prototypes are situated at optimal locations.

As can be seen from the update rule (3), after convergence in the limit $\lambda \to 0$ it holds

$$\alpha_{ij} \longrightarrow \left\{ \begin{array}{ccc} 1/|R_i| & : & v_j \in R_i \\ 0 & : & v_j \notin R_i \end{array} \right. , \text{ because } \left\{ \begin{array}{c} h_\lambda(k_{ij}) = 1 \text{ for } v_j \in R_i \\ h_\lambda(k_{ij}) \to 0 \text{ for } v_j \notin R_i \end{array} \right\},$$

where $R_i = \{v_j \in V : \|w_i - v_j\| \leq \|w_k - v_j\| \text{ for all } k\}$ denotes the receptive field of prototype w_i. That means, in the limit only datapoints from the receptive fields have positive coefficients and equally contribute to the winning prototype that is located in the center of gravity of its receptive field.

A *k-approximation* of an optimal relational prototype w_i is a subset $R' \subseteq R_i$ with $|R'| = \min\{k, |R_i|\}$ such that $\sum_{r' \in R'} \|w_i - r'\|^2$ is minimized. That means, we choose the k nearest points from the receptive field of a prototype as representatives. If there are less than k points in the receptive field, the whole field is taken. This computation can be done in time $\mathcal{O}(|R_i| \cdot k)$. For a set W of relational prototypes, we refer to the set containing a k-approximation for each relational prototype $w_i \in W$ a *k-approximation* of W.

These k-approximations in combination with their corresponding coefficients can be interpreted as a convex-combined point in the relational model, defined just over the points of the k-approximation. Therefore, if merged into the next patch, the number of the prototype coefficients remains limited, and the distances of these approximated prototypes to points of the next patch can be calculated using the original equations. This way, only a fraction of the inter-patch distances needs to be considered.

Construction of Extended Patches Let W_t be a set of optimal relational prototypes gained in a step t. Assume N_t denotes the index set of all points included in the union of a k-approximation of W_t pointing onto elements of the dissimilarity matrix D. The extended patch P_t^* is then characterized by the distance matrix

$$P_t^* = \left(\begin{array}{c|c} d(N_{t-1}) & d(N_{t-1}, P_t) \\ \hline d(N_{t-1}, P_t)^T & P_t \end{array} \right)$$

where

$$d(N_{t-1}) = (d_{uv})_{u,v \,\in\, N_{t-1}} \;\in\; \mathbb{R}^{n_t \times n_t}$$
$$d(N_{t-1}, P_t) = (d_{uv})_{u \,\in\, N_{t-1}, v=(t-1)\cdot p+1,\ldots,t\cdot p} \;\in\; \mathbb{R}^{n_t \times p}$$

denote the inter-distances of points from the k-approximation and the distances between points from the k-approximation and current patch points, respectively. The size n_t is bounded by $|W_t| \cdot k$.

Integrating Multiplicities The original Relational Neural Gas method has to be modified to handle datapoints v_j equipped with multiplicities m_j which are given by the size of the receptive fields divided by k. Incorporating multiplicities into the cost function yields the update rule

$$\bar{\alpha}_{ij} = \frac{m_j \cdot h_\lambda(k_i(v_j))}{\sum_t m_t \cdot h_\lambda(k_i(v_t))}$$

for prototype coefficients. The computation of distances is not changed.

Patch Relational Neural Gas Assembling the pieces, we obtain:

```
Algorithm
Cut the first Patch $P_1$
Apply Relational NG on $P_1$ $\longrightarrow$ Relational prototypes $W_1$
Use $k$-Approximation on $W_1$ $\longrightarrow$ Index set $N_1$
Update Multiplicities $m_j$ according to the receptive fields

Repeat for $t = 2, \ldots, n_p$
   Cut patch $P_t$
   Construct Extended Patch $P_t^*$ using $P_t$ and index set $N_{t-1}$
   Apply modified RNG with Multiplicities $\longrightarrow$ Relational prototypes $W_t$
   Use $k$-Approximation on $W_t$ $\longrightarrow$ Index set $N_t$
   Update Multiplicities $m_j$ according to the receptive fields
Return $k$-approximation of final prototypes $N_{n_p}$
```

Complexity Obviously, the size of extended patches is bounded by the size of the new patch read from the distance matrix and the distances of the at most $k \cdot n$ points representing the n prototypes of the last run by their k approximation. Assume a bounded extended patch size p independent of the number of datapoints, as it would be the case when the patch size is chosen according to memory limitations. The algorithm then works only on $\mathcal{O}(\frac{m}{p} \cdot p^2) = \mathcal{O}(m \cdot p) = \mathcal{O}(m)$ entries of the dissimilarity matrix, compared to $\mathcal{O}(m^2)$ in the original Median NG method. Moreover, the algorithm uses at most $\mathcal{O}(p^2) = \mathit{const}$ entries at a specific point in time.

In case of fixed patch size, also the time complexity is linear, because the Median NG step is $\mathcal{O}(p^2)$ what results in $\mathcal{O}(p^2 \cdot \frac{m}{p}) = \mathcal{O}(p \cdot m) = \mathcal{O}(m)$, an advantage compared to the $\mathcal{O}(m^2)$ time complexity of the original Median NG. Further, the algorithm can be run in a single pass over the data.

These advantages in space and time complexity are obtained by an approximation of the prototypes. As we will see in experiments, this leads only to a small loss in accuracy.

5 Experiments

Practioners often handle huge datasets whose dissimilarities cannot be hold at once within random-access memory due to the sheer amount of data ($\mathcal{O}(m^2)$). At that point, Patch Relational NG comes into play providing a single pass technique based on patches that can be chosen in accordance to the available random-access memory. To show the overall performance of the proposed method, we have chosen some representative dissimilarity datasets. Due to limited computing power and hardware available, the chosen datasets do not represent real-life huge datasets, they should be understood as a proof-of-concept that nevertheless can instantly be transfered to the real problems.

We evaluate the clustering results by means of the classification error for supervised settings, whereby class labels are obtained by posterior labeling of prototypes. Note, however, that the goal of the algorithms is meaningful clustering of data based on a chosen similarity measure and cost function. Hence, the classification error gives only a hint about the quality of the clustering, depending on whether the class labels are compatible to the data clusters and chosen metric or not. We accompany this supervised evaluation be the standard quantization error of the clustering.

For all experiments the initial neighborhood range λ_0 is chosen as $n/2$ with n the number of neurons used. The neighborhood range $\lambda(t)$ is decreased exponentially with the number of adaptation steps t according to $\lambda(t) = \lambda_0 \cdot (0.01/\lambda_0)^{t/t_{\max}}$ (cf. [12]). The value $t_{\max}$ is chosen as the number of epochs.

Synthetic Dataset

To analyze the relation between the number of patches and the quantization error on one hand, and the effect of k-approximation of relational prototypes on the other hand, an artificial dataset from [3] was taken. It consists of 1250 datapoints in the Euclidean plane gained from three Gaussian clusters.

Effect of *k*-Approximation For an empirical study of the effect of k-approximation on the quantization error, we trained 50 neurons with the original Relational NG for 100 epochs, i.e. on average every neuron represents 25 datapoints. On the outcoming relational neurons, k-Approximation for $k = 1, \ldots, 20$ were applied. Figure 1 shows a comparison of the quantization errors yielded with the different approximations to the quantization error gained by the original relational neurons. For each step the average over 10 runs is reported.

As expected, the quantization error decreases with higher numbers k of datapoints used to approximate each relational neuron. Concerning the patch approach, applying a k-approximation to the relational neurons of each patch

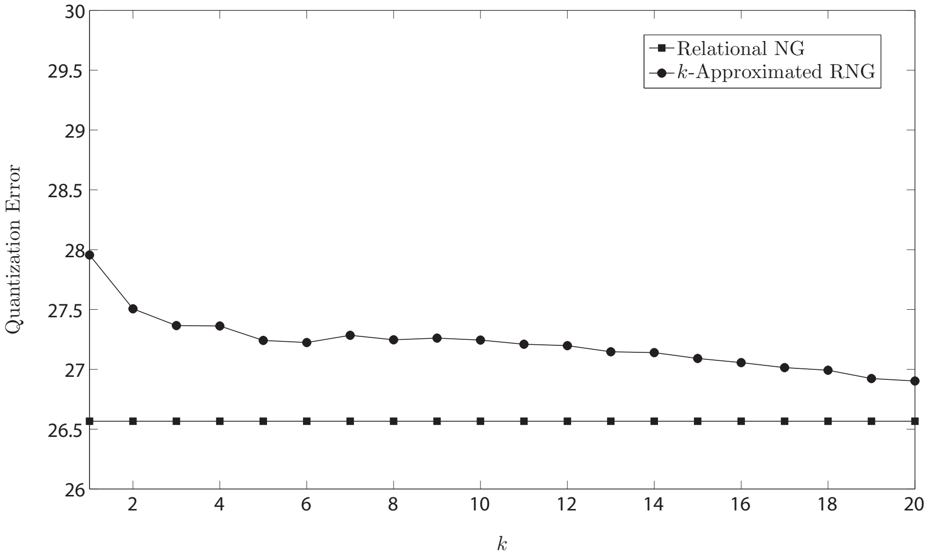

Fig. 1. Quantization error (i.e. $E(W)$ for $\lambda \to \infty$) of original relational neurons compared to different k-approximations on a synthetic dataset

clearly results in a loss of accuracy depending on the choice of parameter k. But as can be seen later on, even with k-approximation the quality of the results is still convincing.

Effect of Patch Sizes Analyzing the relation between the number of patches chosen and the quantization error, we trained median and relational NG with 20 neurons for 50 epochs. The results presented in figure 2 show the quantization error averaged over 10 runs for each number of patches. As expected, the quantization error increases with the number of patches used. But compared to the Median Patch NG approach the presented Patch Relational NG performs very well with only a small loss even for a larger number of patches used.

Chicken Pieces Silhouettes Dataset

The task is to classify 446 silhouettes of chicken pieces into the categories wing, back, drumstick, thigh and back, breast. Data silhouettes are represented as a string of the angles of consecutive tangential pieces of length 20, including appropriate scaling. Strings are compared using a (rotation invariant) edit distance, where insertions/deletions cost 60, and the angle difference is taken otherwise.

For training we used 30 neurons. For Patch Median NG the dataset was divided into 4 patches, i.e. a patch size of around 111 datapoints. The results reported in Table 1 are gained from a repeated 10-fold stratified crossvalidation averaged over 100 repetitions and 100 epochs per run. The k-approximation for Patch Relational NG was done with $k = 3$.

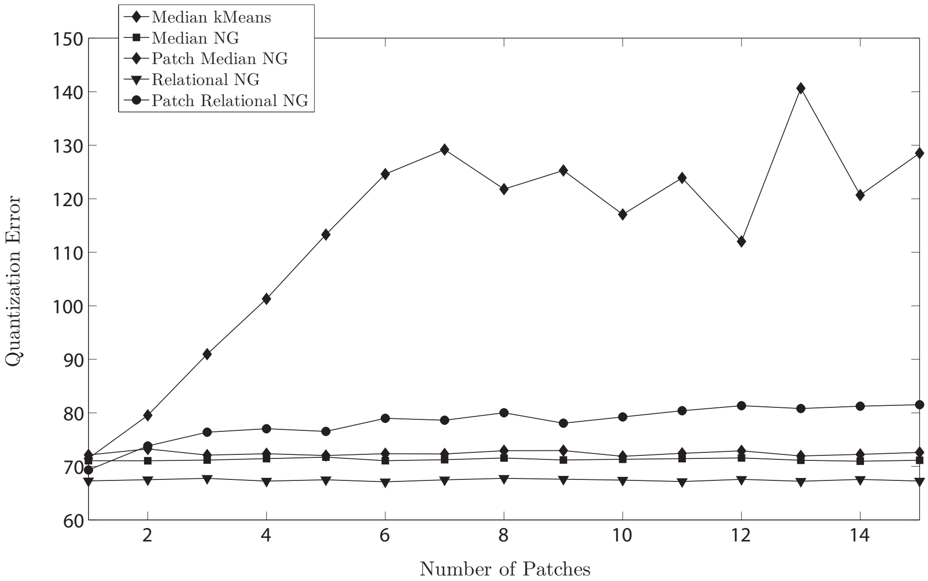

Fig. 2. Quantization error for different patch sizes on a synthetic dataset

Protein Classification

The evolutionary distance of 226 globin proteins is determined by alignment as described in [13]. These samples originate from different protein families: hemoglobin-α, hemoglobin-β, myoglobin, etc. Here, we distinguish five classes as proposed in [6]: HA, HB, MY, GG/GP, and others. Table 2 shows the class distribution of the dataset.

For training we used 20 neurons. For Patch Median NG the dataset was divided into 4 patches, i.e. a patch size of around 57 datapoints. The results reported in Table 3 are gained from a repeated 10-fold stratified crossvalidation averaged over 100 repetitions and 100 epochs per run.

Despite the small size of this dataset – acting more as a proof-of-concept example – the results clearly show a good performance of Patch Median NG. Nevertheless, the price of reduced accuracy is obvious, but faster computation

Accuracy on Chicken Pieces Dataset

	Relational NG	Patch Relational NG	Median Batch NG	Patch Median NG	Median k-Means
Mean	84.7	85.4	66.4	68.8	72.9
StdDev	1.0	1.1	1.9	2.3	1.7

Table 1. Classification accuracy on Chicken Pieces Dataset gained from repeated 10-fold stratified crossvalidation over 100 repetitions, four patches were used.

Class No.	Count	Percentage
HA	72	31.86%
HB	72	31.86%
MY	39	17.26%
GG/GP	30	13.27%
Others	13	5.75%

Table 2. Class Statistics of the Protein Dataset

and less space requirements are gained in return. The k-approximation for Patch Relational NG was done with $k = 3$.

Wisconsin breast cancer

The Wisconsin breast cancer diagnostic database is a standard benchmark set from clinical proteomics [17]. 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.

Dissimilarities were derived by applying the Cosine Measure

$$d_{cos}(v_i, v_j) = 1 - \frac{v_i \cdot v_j}{\|v_i\|_2 \cdot \|v_j\|_2}.$$

We trained 40 neurons for 100 epochs. As result the accuracy on the test set for a repeated 10-fold stratified crossvalidation averaged over 100 runs is reported. The number of patches chosen for Patch Median NG and Patch Relational NG was 5, i.e. around 114 datapoints per patch. The k-approximation for Patch Relational NG was done with $k = 2$.

Also on this dataset, Patch Relational NG acts merely worse than the original Relational NG. Though, the reduction in accuracy is clearly observable.

Chromosome Images Dataset

The Copenhagen chromosomes database is a benchmark from cytogenetics. A set of 4200 human chromosomes from 22 classes (the autosomal chromosomes)

Accuracy on Protein Dataset

	Relational NG	Patch Relational NG	Median Batch NG	Patch Median NG	Median k-Means
Mean	92.62	92.61	79.9	77.7	80.6
StdDev	0.92	0.88	1.5	2.4	1.3

Table 3. Classification accuracy on Protein Dataset gained from repeated 10-fold stratified crossvalidation over 100 repetitions, four patches were used.

Accuracy on Wisconsin Breast Cancer Dataset

	Relational NG	Patch Relational NG	Median Batch NG	Patch Median NG	Median k-Means
Mean	95.0	94.8	94.7	94.4	94.6
StdDev	0.6	0.7	0.7	0.7	0.7

Table 4. Classification accuracy on Wisconsin Breast Cancer Dataset with Cosine Measure gained from repeated 10-fold stratified crossvalidation over 100 repetitions, five patches and a 2-approximation were used.

are represented by the grey levels of their images. These images were transferred to strings representing the profile of the chromosome by the thickness of their silhouettes. Strings were compared using edit distance with substitution costs given by the signed difference of the entries and insertion/deletion costs given by 4.5 [14]. The methods have been trained using 60 neurons for 100 epochs. As result the accuracy on the test set for a repeated 2-fold stratified crossvalidation averaged over 10 runs is reported. The number of patches chosen for Patch Median NG and Patch Relational NG was 10, i.e. around 420 datapoints per patch. The k-approximation for Patch Relational NG was done with $k = 3$.

Also on this dataset, Patch Relational NG acts well. Though, the reduction in accuracy is clearly observable.

6 Summary

In this paper, we proposed a special computation scheme, based on Relational Neural Gas, that allows to process huge dissimilarity datasets by a single pass technique of fixed sized patches. The patch size can be chosen to match the given memory constraints. As explained throughout the paper, the proposed patch version reduces the computation and space complexity with a small loss in accuracy, depending on the patch sizes. We further demonstrated the ability of the proposed method on several representative clustering and classification problems. In all experiments, relational adaptation increased the accuracy of Median clustering.

Accuracy on Copenhagen Chromosome Image Dataset

	Relational NG	Patch Relational NG	Median Batch NG	Patch Median NG	Median k-Means
Mean	89.6	87.0	80.0	67.9	77.1
StdDev	0.6	0.8	1.4	3.1	2.2

Table 5. Classification accuracy on Copenhagen Chromosome Image Dataset gained from repeated 2-fold stratified crossvalidation over 10 repetitions, 10 patches and a 3-approximation were used.

Note that relational and patch optimization are based on a cost function related to NG such that extensions including semisupervised learning and metric adaptation can directly be transferred to this settings. In future work, the method will be applied to more real-world datasets. The patch scheme also opens a way towards parallelizing the method as demonstrated in [2].

References

1. N. Alex, B. Hammer, and F. Klawonn, Single pass clustering for large data sets, *WSOM*, 2007.
2. N. Alex and B. Hammer, Parallelizing single patch pass clustering, submitted to ESANN 2008
3. M. Cottrell, B. Hammer, A. Hasenfuss, and T. Villmann (2006), Batch and median neural gas, *Neural Networks*, 19:762-771.
4. S. Guha, N. Mishra, R. Motwani, L. O'Callaghan (2000). Clustering Data Streams. In *IEEE Symposium on Foundations of Computer Science*, 359-366.
5. S. Guha, R. Rastogi, K. Shim (1998). CURE: an efficient clustering algorithm for large datasets. In *Proceedings of ACM SIGMOD International Conference on Manag ement of Data*, 73-84.
6. B. Haasdonk and C. Bahlmann (2004), Learning with distance substitution kernels, in *Pattern Recognition - Proc. of the 26th DAGM Symposium.*
7. B. Hammer and A. Hasenfuss, Relational Neural Gas. In J. Hertzberg et al., editors, *KI 2007: Advances in Artificial Intelligence*, Lecture Notes in Artifical Intelligence 4667, pages 190-204, Springer, 2007.
8. J.A.Hartigan (1975), Clustering Algorithms, Wiley.
9. R. Jin, A. Goswami, G. Agrawal (to appear). Fast and Exact Out-of-Core and Distributed K-Means Clustering, *Knowledge and Information System.*
10. T. Kohonen (1982), Self-Organized formation of topologically correct feature maps, *Biological Cybernetics*, 43:59-69.
11. T. Kohonen and P. Somervuo (2002), How to make large self-organizing maps for nonvectorial data, *Neural Networks* **15:**945-952.
12. T. Martinetz, S. Berkovich, and K. Schulten (1993). 'Neural gas' network for vector quantization and its application to time series prediction. *IEEE Transactions on Neural Networks*, 4(4):558–569.
13. H. Mevissen and M. Vingron (1996), Quantifying the local reliability of a sequence alignment, *Protein Engineering* **9:**127-132.
14. M. Neuhaus and H. Bunke, Edit distance based kernel functions for structural pattern classification *Pattern Recognition* 39(10):1852-1863, 2006.
15. Y. Prudent, A. Ennaji (2005). An incremental growing neural gas learns topology. IJCNN'05.
16. W. Wang, J. Yang, R.R. Muntz (1997). STING: a statistical information grid approach to spatial data mining. In *Proceedings of the 23rd VLDB Conference*, 186-195.
17. 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.
18. T. Zhang, R. Ramakrishnan, M. Livny (1996). BIRCH: an efficient data clustering method for very large databases. In *Proceedings of the 15th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 103-114.