

Intuitive Clustering of Biological Data

Barbara Hammer¹, Alexander Hasenfuss¹, Frank-Michael Schleif²,
Thomas Villmann², Marc Strickert³, and Udo Seiffert³

Abstract—K-means clustering combines a variety of striking properties because of which it is widely used in applications: training is intuitive and simple, the final classifier represents classes by geometrically meaningful prototypes, and the algorithm is quite powerful compared to more complex alternative clustering algorithms. In this contribution, we focus on extensions which incorporate additional information into the clustering algorithm to achieve a better accuracy: neighborhood cooperation from neural gas, (possibly fuzzy) label information of input data, and general problem-adapted distances instead of the standard Euclidean metric. These extensions can be formulated in a simple general framework by means of a cost function. We demonstrate the ability of these variants on several representative clustering problems from computational biology.

I. INTRODUCTION

Prototype-based or k-means clustering is widely used in problems of computational biology such as gene expression analysis [16], [20], [23]. The algorithm combines several striking properties: it is fast, intuitive and very easy to implement, the resulting classes are represented by their means and simple to interpret, and, despite its simplicity, the algorithm is quite powerful. However, the algorithm has several drawbacks when applied to typical problems in computational biology: it optimizes the quantization error by means of a simple batch optimization scheme similar to EM (Expectation Maximization) algorithms [5]. It is well known that these methods easily get stuck in local optima of the cost function such that multiple restarts or computationally complex metaheuristics such as genetic algorithms or simulated/deterministic annealing have to be used [16]. Further, in particular for high dimensional or noisy data, minimization of the quantization error does not yield meaningful clusters, rather, additional information such as cluster labels or correlation should be taken into account [12], [13]. In addition, it depends heavily on the underlying Euclidean metric and yields inappropriate results if the standard Euclidean metric is not suited (e.g. for gene expression data where up- and down-regulations are more important than the absolute values [6], [25]) or not applicable at all (e.g. for protein sequences of different length [19]).

Various approaches to overcome these problems have been proposed in the literature: clustering for general proximity data [7], [22], clustering with auxiliary information [13], clustering using annealing schemes or metaheuristics [7], [16], clustering with adapted metric [1] or alternative general approaches, see e.g. [29] for an overview. However, these methods do not share the simplicity and easy interpretability of prototype-based clustering, often requiring quite complex optimization schemes, such that they do not approximate its popularity.

Recently, various clustering algorithms which partially incorporate these issues and which are based on a cost function similar to the quantization error have been proposed in the context of prototype-based clustering: neighborhood

cooperation of neural gas networks [4], [18], [26] is included to avoid local optima; (potentially fuzzy) label information can be incorporated into the clustering by means of an extension of the cost function [26], [27]; the so-called generalized median allows to apply batch clustering to general distance matrices [4], [15]. Here, we systematically combine and analyze these ideas and obtain a family of clustering algorithms which can cope with the above-mentioned problems, thereby maintaining the intuitivity of prototype-based clustering. We demonstrate the effect of the different variants on several typical classification problems from computational biology.

II. BASIC CLUSTERING ALGORITHM

Assume data vectors $\mathbf{v} \in \mathbb{R}^d$ are given as stimuli, distributed according to an underlying probability distribution $P(\mathbf{v})$. The aim of prototype-based unsupervised clustering is to find a number of prototypes or weight vectors $\mathbf{w}_i \in \mathbb{R}^d$, $i = 1, \dots, k$ representing the data points faithfully, e.g. measured in terms of the average deviation of a data point from its respective closest prototype. The objective of clustering is a minimization of the quantization error

$$E_{\text{KM}}(W) = \frac{1}{2} \sum_{j=1}^k \int \chi_j(\mathbf{v}) \cdot (\mathbf{v} - \mathbf{w}_j)^2 P(\mathbf{v}) d\mathbf{v}$$

where $\chi_j(\mathbf{v}) \in \{0, 1\}$ equals 1 iff \mathbf{w}_j is the prototype located closest to \mathbf{v} in the Euclidean norm. For a given finite set of training data $\mathbf{v}_1, \dots, \mathbf{v}_n$, sampled according to P , the quantization error can be optimized in a batch-optimization scheme by determining in turn optimum assignments of data points to prototypes and optimum prototype locations for the assigned data points, i.e. the two following steps are repeated until convergence:

- 1) Given fixed \mathbf{w}_j , determine $\chi_{ij} = 1$ if \mathbf{w}_j is closest to \mathbf{v}_i , and 0 otherwise.
- 2) Given fixed χ_{ij} , determine $\mathbf{w}_j = \frac{\sum_{i=1}^n \chi_{ij} \mathbf{v}_i}{\sum_{i=1}^n \chi_{ij}}$ as center of gravity of the data points assigned to \mathbf{w}_j .

These steps are obtained when optimizing the cost function E_{KM} for a given finite set of data points where the assignments $\chi_j(\mathbf{v}_i)$ are taken as hidden variables for fixed \mathbf{w}_j resp. fixed assignments. It has been shown in [2] that this algorithm converges in a finite number of steps and it can be interpreted as a (fast) Newton optimization scheme.

A. Label information

Assume additional information in the form of (possibly fuzzy) class labels is available which should be taken into account when building the clusters. That means, for every data point \mathbf{v}_i a class label $y_i \in \mathbb{R}^d$, d being the number of classes, is available. Crisp labels correspond to unit vectors, probabilistic interpretation requires that the components of each y_i add up to 1. Here, general possibilistic labels can be dealt with, i.e. no conditions have to be met for y_i . It has been proposed in [9], [26], [27] to incorporate this information into clustering by extending the respective cost function by a term which deals with the correctness of labels within

1. Clausthal University of Technology, Institute of Informatics, Clausthal-Zellerfeld, Germany; 2. University of Leipzig, Clinic for Psychotherapy, Leipzig, Germany; 3. Pattern Recognition Group, Leibniz Institute of Plant Genetics and Crop Plant Research, Gatersleben, Germany)

a cluster. Here, we transfer the idea of [9]: assume each prototype is equipped with a label $Y_i \in \mathbb{R}^d$ which represents the mean fuzzy label of cluster i and which is automatically determined during training. The cost function of supervised-k-means becomes $E_{\text{SKM}}(W, Y) =$

$$(1 - \alpha) \cdot \frac{1}{2} \cdot \sum_{j=1}^k \int \chi_j(\mathbf{v}, y) \cdot (\mathbf{v} - \mathbf{w}_j)^2 P(\mathbf{v}) d\mathbf{v} \\ + \alpha \cdot \frac{1}{2} \cdot \sum_{j=1}^k \int \chi_j(\mathbf{v}, y) \cdot (y - Y_j)^2 P(\mathbf{v}) d\mathbf{v}$$

where $\chi_j(\mathbf{v}, y) = 1$ if $(1 - \alpha) \cdot (\mathbf{v} - \mathbf{w}_j)^2 + \alpha \cdot (y - Y_j)^2 < (1 - \alpha) \cdot (\mathbf{v} - \mathbf{w}_l)^2 + \alpha \cdot (y - Y_l)^2$ for all $l \neq j$, and it is 0, otherwise. That means, the prototype \mathbf{w}_j with class label Y_j is closest to the data point \mathbf{v} and its label y in the Euclidean norm enhanced by the label information. Thereby, $\alpha \in [0, 1]$ constitutes a weighting of the two objectives, label learning and a distribution of prototypes among the data. The quantization error is recovered for $\alpha = 0$.

This cost function can be optimized in batch mode for a given finite set of data. Optimum values of W and Y , given fixed assignments of the data points, and, in turn, optimum assignments, given fixed W and Y can be directly computed. As a result, the following two steps are iterated until convergence:

- 1) For given W, Y , set $\chi_{ij} = 1$ if $(1 - \alpha) \cdot (\mathbf{v}_i - \mathbf{w}_j)^2 + \alpha \cdot (y_i - Y_j)^2 < (1 - \alpha) \cdot (\mathbf{v}_i - \mathbf{w}_l)^2 + \alpha \cdot (y_i - Y_l)^2$ for all $l \neq j$, and it is 0, otherwise.
- 2) For fixed χ_{ij} , set $\mathbf{w}_j = \sum_i \chi_{ij} \cdot \mathbf{v}_i / \sum_i \chi_{ij}$, and $Y_j = \sum_i \chi_{ij} \cdot y_i / \sum_i \chi_{ij}$.

Note that the assignments of the receptive fields depend on the closeness of the prototype as well as the correctness of its class label, such that cluster borders tend to follow the class labels depending on the choice of α . The convergence proof provided in [9] applies to this scenario: the algorithm converges in a finite number of steps, whereby the solution is a (local) optimum of the cost function under mild conditions on the output. Further, the scheme can be interpreted as a Newton optimization w.r.t. prototype and label locations, as can be seen as follows: the updates of prototypes and labels can be written in the form

$$\Delta \mathbf{w}_j = \frac{\sum_i \chi_{ij} \cdot (\mathbf{v}_i - \mathbf{w}_j)}{\sum_i \chi_{ij}}$$

and

$$\Delta Y_j = \frac{\sum_i \chi_{ij} \cdot (y_i - Y_j)}{\sum_i \chi_{ij}}.$$

Newton's method for an optimization of the cost function yields

$$\Delta(\mathbf{w}_j, Y_j) = -J \cdot H^{-1}$$

whereby J is the Jacobian of the cost function w.r.t. \mathbf{w}_j and Y_j and H the Hessian matrix. Since χ_{ij} is locally constant, we obtain up to sets of measure zero the Jacobian J with entries

$$((1 - \alpha) \cdot \sum_i \chi_{ij} (\mathbf{w}_j - \mathbf{v}_i), \alpha \cdot \sum_i \chi_{ij} (Y_j - y_i))_j$$

and the Hessian with diagonal entries $(1 - \alpha) \cdot \sum_i \chi_{ij}$ for the components corresponding to \mathbf{w}_j and $\alpha \cdot \sum_i \chi_{ij}$ for the components corresponding to Y_j , thus the above formulas result.

B. Neighborhood cooperation

There exists a variety of methods which include neighborhood cooperation into clustering, such as the self-organizing map (SOM) or neural gas (NG) [14], [18]. Due to its fixed prior lattice, usually a two-dimensional Euclidean grid, SOM is widely used for simultaneous data clustering and visualization. However, the topology puts a bias towards clustering which is misleading if the inherent data topology is not of the same dimensionality. Neural gas has the benefit that it learns a data optimum topology. The cost function of NG is given by

$$E_{\text{BNG}}(W) = \frac{1}{2C(\lambda)} \cdot \sum_{j=1}^k \int h_\lambda(k_j(\vec{v}, W)) \cdot (\vec{v} - \vec{w}_j)^2 P(d\vec{x})$$

where $k_j(\vec{v}, W) = |\{\vec{w}_l \mid (\vec{v} - \vec{w}_l)^2 < (\vec{v} - \vec{w}_j)^2\}|$ is the rank of the prototypes sorted according to the distances, $h_\lambda(t) = \exp(-t/\lambda)$ is a Gaussian shaped curve with neighborhood range $\lambda > 0$, and $C(\lambda)$ is the constant $\sum_{j=1}^k h_\lambda(k_j)$. In the limit of small neighborhood cooperation, i.e. $\lambda \rightarrow 0$, the standard quantization error of k-means is recovered. For $\lambda > 0$, the multimodal quantization error is smoothed and it becomes unimodal for $\lambda \rightarrow \infty$. On the one hand, this neighborhood cooperation avoids that prototypes get trapped in local optima. On the other hand, the use of the ranks in NG allows an intrinsic magnification of differences of the distances, which, particularly in high dimensions, tend to be very similar due to the curse of dimensionality. Therefore, a better convergence can be observed for NG than for k-means clustering. NG is usually trained with decreasing neighborhood cooperation such that good optima of k-means can robustly be found by this method. Usually, NG is trained in an online mode. For a given training set, alternative batch optimization can be applied as derived in the article [4] from the NG cost function together with a convergence proof similar to [2]. We refer to this batch optimization scheme by BNG.

Again supervised label information can be incorporated, yielding the cost function $E_{\text{SBNG}}(W, Y) =$

$$(1 - \alpha) \cdot \frac{1}{2C(\lambda)} \cdot \sum_{j=1}^k \int h_\lambda(k_j(\mathbf{v}, y, W, Y)) \cdot (\mathbf{v} - \mathbf{w}_j)^2 P(\mathbf{v}) d\mathbf{v} \\ + \alpha \cdot \frac{1}{2C(\lambda)} \cdot \sum_{j=1}^k \int h_\lambda(k_j(\mathbf{v}, y, W, Y)) \cdot (y - Y_j)^2 P(\mathbf{v}) d\mathbf{v}$$

where $k_j(\mathbf{v}, y, W, Y) = |\{\mathbf{w}_l \mid (1 - \alpha) \cdot (\mathbf{v} - \mathbf{w}_l)^2 + \alpha \cdot (y - Y_l)^2 < (1 - \alpha) \cdot (\mathbf{v} - \mathbf{w}_j)^2 + \alpha \cdot (y - Y_j)^2\}|$ denotes the rank of prototype i measured according to the closeness of the current data point and the prototype weight and labeling. For a given finite set of training data, batch optimization can be performed by iterating the following two optimization steps until convergence:

- 1) For given W, Y , set $k_{ij} = |\{\mathbf{w}_l \mid (1 - \alpha) \cdot (\mathbf{v}_i - \mathbf{w}_l)^2 + \alpha \cdot (y_i - Y_l)^2 \leq (1 - \alpha) \cdot (\mathbf{v}_i - \mathbf{w}_j)^2 + \alpha \cdot (y_i - Y_j)^2\}|$ as the rank of prototype j given \mathbf{v}_i .
- 2) For fixed k_{ij} , set $\mathbf{w}_j = \sum_i h_\lambda(k_{ij}) \cdot \mathbf{v}_i / \sum_i h_\lambda(k_{ij})$, and $Y_j = \sum_i h_\lambda(k_{ij}) \cdot y_i / \sum_i h_\lambda(k_{ij})$.

As before, convergence can be guaranteed [9], and the unsupervised versions are recovered for $\alpha = 0$. One can see as beforehand that this method can be interpreted as Newton optimization. Note that this update scheme maintains the intuitive character of k-means: the main differences consist in an extension of the metric by label information and an

incorporation of all training points according to their rank to determine cluster means and the averaged cluster label, respectively.

C. Median clustering

Often, data are not embedded in a Euclidean vector space or the Euclidean metric is not appropriate for comparison, such that the standard clustering versions cannot be applied. Instead, pairwise proximities $d_{ij}^2 = d(\mathbf{v}_i, \mathbf{v}_j)$ are available, which can stem from an arbitrary similarity measure or even an experimental evaluation. Thereby, there are no assumptions on these values such as symmetry or positive definiteness. There exist several approaches in the literature to extend vectorial clustering and classification methods to this general setting such as supervised learning vector quantization for general metrics [10], [24], or self-organizing clustering for proximity data [22], [7]. However, these methods require either specific properties and adaptations of the algorithm according to the metric, or they do no longer rely on a simple cost function similar to k-means. Here, we use the idea presented in [4], [15]: the mean is substituted by the generalized median. This procedure has the benefit that it can be immediately applied to any given distance matrix, and it maintains the simplicity of standard k-means.

The main problem which prevents the applicability of k-means consists in the fact that the prototype locations cannot be adapted within a continuous vector space and direct analytical optimization of the cost term given fixed assignments is not possible. In this case, however, a discrete adaptation of the prototypes within the locations given by the data space is possible, i.e. we choose $w_j = \mathbf{v}_{l_j}$ for some l_j .

Assume values $d_{ij}^2 = d(\mathbf{v}_i, \mathbf{v}_j)$ are available. Then we can formalize the cost function of k-means for the given (finite) data set in the following form for generalized median clustering

$$\hat{E}_{\text{KMMedian}}(W) = \frac{1}{2} \cdot \sum_{i=1}^n \sum_{j=1}^k \chi_j(\mathbf{v}_i) \cdot d_{il_j}^2$$

where $\mathbf{w}_j = \mathbf{v}_{l_j}$ characterizes the position of prototype j . As a consequence, batch optimization restricts possible prototype locations to the set of given training data. We obtain by iterative optimization of this cost function the batch steps

- 1) For given W where $\mathbf{w}_j = \mathbf{v}_{l_j}$, set $\chi_{ij} \in \{0, 1\}$ to 1 iff $d_{il_j}^2$ is minimum.
- 2) For fixed χ_{ij} , set $\mathbf{w}_j = \mathbf{v}_{l_j}$ for which $\sum_i \chi_{ij} \cdot d_{il_j}^2$ is minimum.

Thereby, the minimum in step (2) is determined by extensive search. Optimization techniques either by exact reformulation of the procedure as presented in [3] for median SOM, or by heuristic optimization e.g. restricting the set of potential candidate locations to data points assigned to the respective prototype allow to speed up this procedure.

The same idea can be transferred to the versions incorporating supervision and neighborhood cooperation. Supervised median NG, for example, is described by the cost term

$$\hat{E}_{\text{SBNGMedian}}(W, Y) =$$

$$(1 - \alpha) \cdot \frac{1}{2C(\lambda)} \cdot \sum_{i=1}^n \sum_{j=1}^k h_\lambda(k_j(\mathbf{v}_i, y_i, W, Y)) \cdot d_{il_j}^2$$

$$+ \alpha \cdot \frac{1}{2C(\lambda)} \cdot \sum_{i=1}^n \sum_{j=1}^k h_\lambda(k_j(\mathbf{v}_i, y_i, W, Y)) \cdot (y_i - Y_j)^2$$

where $\mathbf{w}_j = \mathbf{v}_{l_j}$ and $k_j(\mathbf{v}_i, y_i, W, Y) = |\{\mathbf{w}_l \mid (1 - \alpha) \cdot d_{il_l}^2 + \alpha(y_i - Y_l)^2 < (1 - \alpha) \cdot d_{il_j}^2 + \alpha(y_i - Y_j)^2\}|$. Batch optimization yields

- 1) For given W, Y , set $k_{ij} = |\{\mathbf{w}_l \mid (1 - \alpha) \cdot d_{il_l}^2 + \alpha \cdot (y_i - Y_l)^2 \leq (1 - \alpha) \cdot d_{il_j}^2 + \alpha \cdot (y_i - Y_j)^2\}|$ as the rank of prototype j given \mathbf{v}_i .
- 2) For fixed k_{ij} , set $\mathbf{w}_j = \mathbf{v}_{l_j}$ for which $\sum_{i,j} h_\lambda(k_{ij}) \cdot d_{il_j}^2$ is minimum and $Y_j = \sum_i h_\lambda(k_{ij}) \cdot y_i / \sum_i h_\lambda(k_{ij})$.

Again, the minimum in step (2) is determined by extensive search, which can be accelerated by a restriction of potential candidates and the sum to data points assigned to the respective prototype as first or second winner. Supervised median k-means clustering (SKMMedian) is obtained for $\sigma \rightarrow 0$. Unsupervised median NG (BNGMedian) is obtained for $\alpha = 0$. In analogy to [4], [9], convergence of the algorithm in a finite number of steps can be proved: the general convergence prove provided in [9] also applies to this discrete scenario.

Note that the distance of the labels is still the Euclidean metric. Obviously, this could also be substituted by an alternative distance measure such as the cross-entropy.

D. Classification

We will test the suitability of the algorithms to infer meaningful clusters by applying the methods to several classification tasks where explicit label information of the data points is available. We evaluate the resulting clusters in their capability of finding clusters which respect the class labels. This can be evaluated by the classification error induced by the clustering on an independent test set not used for training.

Note that any clustering induces a classification by posterior assignment of a class label to the prototypes. This class label can be determined as the majority of labels of data points assigned to the prototype. This yields an optimum assignment for the labels once the receptive fields are fixed. We will use this labeling for all fully unsupervised methods. We can also use it for the variants incorporating label information; we will refer to these posterior labeling of SKM, SKMMedian, SBNG, and SBNGMedian by SKM⁺, SKMMedian⁺, SBNG⁺, and SBNGMedian⁺.

The incorporation of label information also leads to a direct classification determined by the trained labels Y_i . For crisp classification, we can thereby take the majority component of Y_i as the output label of \mathbf{w}_i . We will refer to this labeling by the standard variants SKM, SKMMedian, SBNG, and SBNGMedian. Posterior labeling of prototypes assigns optimum labels to the prototypes with respect to the Euclidean distance, therefore, the two version SKM and SKM⁺ (resp. SKMMedian and SKMMedian⁺, ...) do not differ if the methods did already converge. Depending on the size of the data set, this need not be the case such that SKM⁺ can yield better results than SKM.

III. EXPERIMENTS

Typical data sets in computational biology show different characteristics depending on the concrete application area: often, data are described by feature vectors such as real-valued measurements for medical diagnosis or global image characteristics for biomedical image processing. Since feature vectors are included in the Euclidean space, standard Euclidean clustering can be used. For biomedical images, alternative characteristics can be found sometimes: characteristic shapes (described e.g. by a sequence of simple form elements such as line segments) or a decomposition of the image into homogeneous regions and their mutual relation (i.e. a graph or tree representation). These latter representations cannot be compared by the Euclidean distance, rather, Hausdorff distances, some form of alignment, or general tree or graph metrics have to be used. Symbolic sequence data of different length constitute another very popular data structure in computational biology, covering genome or protein sequences. Usually, distances of sequences are determined by appropriate alignment recovering evolutionary distances. Another very widespread data type is constituted by (micro- or macro-) array-data characterizing e.g. gene expression patterns. Although these data are contained in the Euclidean vector space, the standard Euclidean metric is not the best choice to compare two patterns; rather, the overall shape of the pattern and the expressed positions of up- and down-regulation are important.

We test the clustering methods proposed in this article on a representative choice of data sets which cover these different aspects: the Wisconsin breast cancer data set, a benchmark from clinical proteomics where data are feature encoded; the Copenhagen chromosomes data set, a benchmark of images from cytogenetics which are represented by strings of different length characterizing the shape of the images; a popular protein data set where pairwise distances are determined to estimate the evolutionary distance; and, finally, microarray data of gene expression experiments. All data sets are labeled such that an evaluation of the clustering by means of the classification error is possible. 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. For simulation the data set was randomly split 50% / 50% into a training set and a test set with fair label distribution.

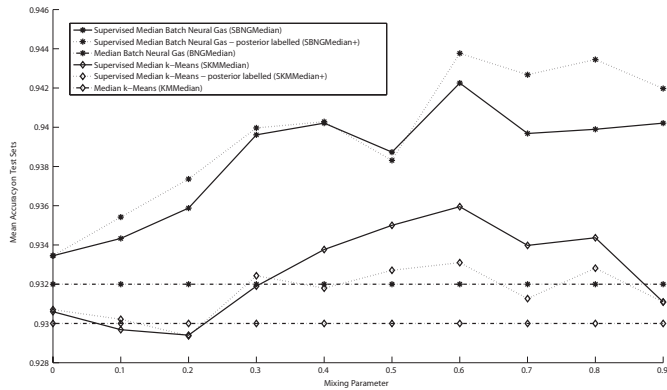


Fig. 1. Classification accuracy (in %) obtained by the different median clustering algorithms for α ranging from 0.1 to 0.9 on the Wisconsin breast cancer dataset.

Wisconsin breast cancer

The Wisconsin breast cancer diagnostic database is a standard benchmark set from clinical proteomics [28]. 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.

The data set is contained in the Euclidean space such that we can compare all clustering versions as introduced above for this data set using the Euclidean metric. We train 40 neurons using 200 epochs. The dataset is z-transformed and randomly split into two halves for each run. The result on the test set averaged over 100 runs is reported. The result for different choices of α is depicted in Fig. 1 for median clustering. For the standard (non-median) Euclidean versions reported in [9]: we obtain a test set accuracy of 0.957 for the supervised version and 0.935 for the unsupervised version, both setting $\alpha = 0.9$ which is optimum for these cases. Results for k-means yield an accuracy 0.938 for standard (unsupervised) k-means resp. 0.941 for supervised k-means. Obviously, there are only minor, though significant differences of the results of the different clustering variants on this data set: incorporation of neighborhood allows to improve k-means, incorporation of label information allows to improve fully unsupervised clustering. As expected, Euclidean clustering is superior to median versions (using the Euclidean norm) because the number of possible prototype locations is reduced for median clustering. However, the difference is only 1.3%, which is quite remarkable in consideration of the comparably small data set, thus dramatically reduced flexibility of prototype locations.

The article [28] reports a test set accuracy of 97.5% using 10-fold cross-validation. Thereby, a problem adapted classification method was used (a large margin linear classifier including feature selection). This differs from our best classification result by 1.8%. Thereby, the goal of our approach is a faithful prototype-based representation of data, such that the result is remarkable.

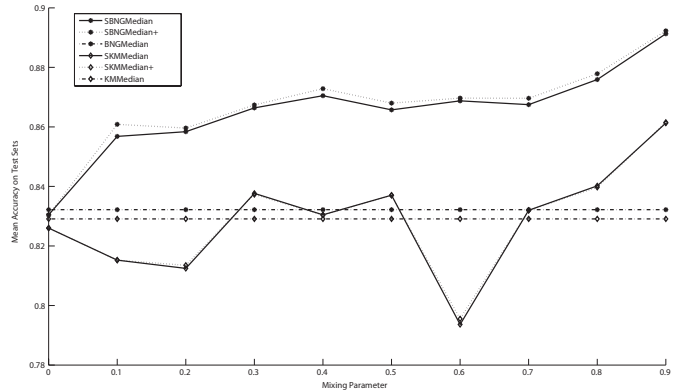


Fig. 2. Results of the methods for the chromosomes database and varying mixing parameter α .

Chromosomes

The Copenhagen chromosomes database is a benchmark from cytogenetics [17]. 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 is non-Euclidean, consisting 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 [11], [21]. 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 [21].

The algorithms have been run using 100 neurons and 100 epochs per run. The results for different mixing parameters α can be seen in Fig. 2. The reported results consist of the test set accuracy averaged over 10 runs. As can be seen, supervised median neural gas achieves an accuracy of 0.89 for $\alpha = 0.9$. This improves by 6% compared to median k-means. A larger number of prototypes allows to further improve this result: 500 neurons yield an accuracy of 0.93 for supervised median neural gas clustering and 0.91 for supervised median k-means clustering, both taken for $\alpha = 0.9$. This is already close to the results of k-nearest neighbor classification which uses all points of the training set for classification. 12-nearest neighbor with the standard edit distance yields an accuracy 0.944 as reported in [11] whereas more compact classifiers such as feedforward networks or hidden Markov models only achieve an accuracy less than 0.91, quite close to our results for only 100 prototypes as shown in Fig. 2.

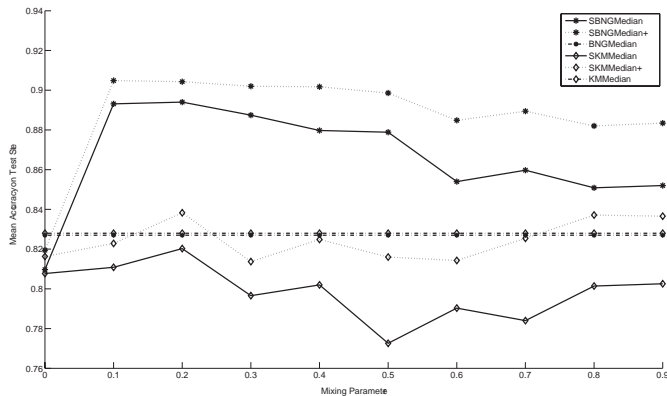


Fig. 3. Results of the methods for the protein database using alignment distance and varying mixing parameter α .

Proteins

The evolutionary distance of 226 globin proteins is determined by alignment as described in [19]. These samples originate from different protein families: hemoglobin- α , hemoglobin- β , myoglobin, etc. Here, we distinguish five classes as proposed in [8]: HA, HB, MY, GG/GP, and others.

We use 30 neurons and 300 epochs per run. The accuracy on the test set averaged over 50 runs is reported for different values α in Fig. 3. Here, the optimum can be observed for supervised median neural gas and $\alpha \in [0.1, 0.5]$, indicating that the statistics of the inputs guides the way towards a good classification accuracy. However, an integration of the labels with small mixing parameter improves the accuracy by nearly 10% compared to fully unsupervised clustering. As beforehand, integration of neighborhood cooperation into k-means is well suited in this scenario. Unlike the results reported in [8] 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,

[8] reports errors which approximately add up to 0.4 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; [8] reports an accumulated leave-one-out error of 0.65 for 1-nearest neighbor) which is comparable to our (clustering) results.

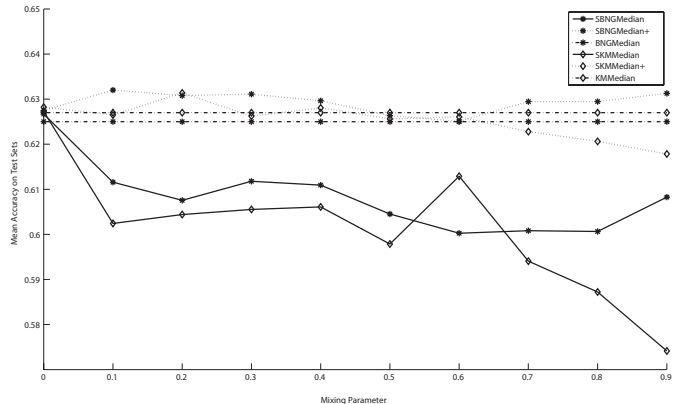


Fig. 4. Results of the median versions for the gene expression database using Pearson correlation and varying mixing parameter α .

Gene expression

The data consist of expression patterns of 1421 genes from filial tissue of barley seeds during 7 developmental stages 0, 2, ..., and 12 days after flowering obtained from macroarrays. Each experiment has been repeated twice for control purpose. A labeling indicating which of these two series data come from is available. As pointed out in [24], a clear experimental bias can be detected which can be tracked to a slight shift in assigning developmental stages in the two sets of independent experiments. This can be detected using e.g. supervised classification methods which report a classification accuracy which clearly deviates from a random classification of 50% for unbiased examples [24].

Data are Euclidean, thus the standard Euclidean distance can be used. However, for gene expression data the correlation of up- and down-regulation is much more important than the absolute size. Therefore, various different similarity measures have been proposed in [24] which include the Pearson correlation of two gene expression series. [24] uses generalized relevance learning vector quantization, a supervised metric-based large-margin classifier which can be adapted to the correlation as distance function. [24] reports a classification accuracy of nearly 50% for the standard Euclidean metric, i.e. a random classification. The similarity $d(\mathbf{v}_i, \mathbf{v}_j) = 1/C(\mathbf{v}_i, \mathbf{v}_j)^4$, $C(\mathbf{v}_i, \mathbf{v}_j)$ being the Pearson correlation coefficient, yields to a test set accuracy 64.95%, which indicates a clear bias of the experiments.

For clustering, we use 15 neurons and 150 epochs. Training and test set are once again randomly split into halves and the average over 25 runs is reported. On the one hand, we use the standard Euclidean metric, which yields an accuracy of about 50% just as reported in [24], i.e. no learning takes place. On the other hand, we train median clustering based on the metric $d(\mathbf{v}_i, \mathbf{v}_j) = 1/C(\mathbf{v}_i, \mathbf{v}_j)^2$, $C(\mathbf{v}_i, \mathbf{v}_j)$ being the Pearson correlation coefficient. Results of these runs are reported in Fig. 4. Clearly, results better than random guess can be obtained, whereby the best result 63.4% approximates the result of the (much more flexible and fully supervised) generalized learning vector quantization. Data are noisy,

i.e. class borders are not well pronounced such that the supervised versions do not yield to convergent labels in only 150 epochs (which is clearly visible because of the difference of the classification given by the trained labels and posterior labeling for the supervised versions). In addition, the supervised versions (with posterior labeling) are only marginally better compared to unsupervised versions, and there is only a small difference between k-means and the version which includes neighborhood cooperation. However, the results clearly emphasize the necessity of choosing an appropriate metric, which becomes possible by substituting standard Euclidean k-means and variants thereof by their median versions.

IV. CONCLUSIONS AND FUTURE WORK

We have presented simple extensions of prototype-based clustering to label information, neighborhood cooperation, and general similarity measures and we have compared the methods on representative biological data sets. The extensions maintain the simplicity, while showing better accuracy due to the avoidance of local optima and a better adaptation of the class borders to additional label information. In all experiments, the classification accuracy of the latter algorithms could outperform simple k-means and, in some examples, it could approximate the results obtained by purely supervised classification methods. We would like to point out that the substitution of class centers by the generalized median allows a formulation of clustering for general similarity measures obtained e.g. by alignment or alternative biologically meaningful consideration such that it opens the way towards a simple and powerful prototype-based clustering for general non-vectorial data. Depending on the problem at hand and the chosen similarity measure, these median versions achieve quite remarkable results.

As already mentioned, apart from a substitution of the metric to compute the similarity of clustering data vectors, also the metric determining label similarities could be chosen in an appropriate problem-dependent way. So far we have restricted simulations to the Euclidean metric for class labels which has the benefit that simple analytic solutions for optimum positions can be found. For probabilistic fuzzy labels, a distance measure such as the cross-entropy would be suitable. However, it is not clear whether this choice yields to an explicit analytic expression for optimum values Y_i .

Another point of interest concerns the updates of prototypes for median clustering. So far, we have restricted prototype positions to discrete locations of the input space, which are determined by extensive search. The optimization can be easily accelerated by restricting the possible locations to input data assigned to the prototype as winner (and second winner, if the neighborhood is taken into account). This leads to a considerable speedup while hardly effecting the accuracy. Details will be investigated in future research.

ACKNOWLEDGMENT

This work was supported by the German Ministry of Education and Research (BMBF) by the grants 0312706A and 0313115.

REFERENCES

[1] S.H. Al-Harbi and V.J. Rayward-Smith (2003), The use of a supervised k-means algorithm on real-valued data with applications in health, in P.W.H.Chung, C.J.Hinde, A.All (eds.), *IEA/AIE 2003*, LNCS 2718, p. 575-581.

[2] L. Bottou and Y. Bengio (1995), Convergence properties of the k-means algorithm, *NIPS*.

[3] 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*, p. 561-568.

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

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin (1977), Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statistical Society – Series B (Methodological)* **39**(1):1-38.

[6] K. Fundel, R. Küffner, T. Aigner, and R. Zimmer (2005), Data processing effects on the interpretation of microarray gene expression experiments, in A.Torda, S.Kurtz, M.Rarey (eds.): *German Conference on Bioinformatics (GCB) 2005*, GI Lecture Notes in Informatics, p. 77-91.

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

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

[9] 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, p. 33-45.

[10] B. Hammer, M. Strickert, and T. Villmann (2005), Supervised neural gas with general similarity measure, *Neural Processing Letters* **21**(1):21-44.

[11] 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.

[12] 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. Schweitzer (eds.), Springer, p. 75-92.

[13] 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.

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

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

[16] Y. Lu, S. Lu, F. Fotouhi, Y. Deng and S.J. Brown (2004), Incremental genetic K-means algorithm and its application in gene expression data analysis, *BMC Bioinformatics* **5**:172.

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

[18] 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.

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

[20] R. Nagarajan and M. Upreti (2006), Correlation Statistics for cDNA Microarray Image Analysis, *Computational Biology and Bioinformatics*, **3**(3):232-238.

[21] M. Neuhaus and H. Bunke (2006), Edit distance based kernel functions for structural pattern classification, to appear in *Pattern Recognition*.

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

[23] R. Shamir and R. Sharan (2001), Approaches to clustering gene expression data, In *Current Topics in Computational Biology*, T. Jiang, T. Smith, Y. Xu, and M.Q. Zhang (eds.), MIT press.

[24] M. Strickert and U. Seiffert and N. Sreenivasulu and W. Weschke and T. Villmann and B. Hammer (2006), Generalized relevance LVQ (GRLVQ) with correlation measures for gene expression analysis, *Neurocomputing* **69**(6-7):651-659.

[25] Y.-H. Taguchi and Y. Oono (2004), Relational patterns of gene expression via non-metric multidimensional scaling analysis, *Bioinformatics* **21**(6):730-740.

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

[27] T. Villmann, U. Seiffert, F.-M. Schleif, C. Brüß, T. Geweniger and B. Hammer (2006), Fuzzy Labeled Self-Organizing Map with Label-Adjusted Prototypes, In *Proceedings of Conference Artificial Neural Networks in Pattern Recognition (ANNPR) 2006*, F. Schwenker (ed.), Springer, p. 46-56.

[28] 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.

[29] S. Zhong and J. Ghosh (2003), A unified framework for model-based clustering, *Journal of Machine Learning Research* **4**:1001-1037.