Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2013, Article ID 165248, 10 pages
http://dx.doi.org/10.1155/2013/165248

Research Article

Clustering by Fuzzy Neural Gas and Evaluation of Fuzzy Clusters

Tina Geweniger, Lydia Fischer, Marika Kaden, Mandy Lange, and Thomas Villmann

Computational Intelligence Group, University of Applied Sciences Mittweida, Technikumplatz 17, 09648 Mittweida, Germany

Correspondence should be addressed to Tina Geweniger; tina@geweniger.org

Received 31 May 2013; Accepted 3 October 2013

Academic Editor: Christian W. Dawson

We consider some modifications of the neural gas algorithm. First, fuzzy assignments as known from fuzzy c-means and neighborhood cooperativeness as known from self-organizing maps and neural gas are combined to obtain a basic Fuzzy Neural Gas. Further, a kernel variant and a simulated annealing approach are derived. Finally, we introduce a fuzzy extension of the ConnIndex to obtain an evaluation measure for clusterings based on fuzzy vector quantization.

1. Introduction

Prototype based vector quantization (VQ) is an approved method to cluster and compress very large data sets. *Prototype based* implies that the data are represented by a much smaller number of prototypes. Famous methods are c-means [1], self-organizing maps (SOM) [2], and neural gas (NG) [3]. These methods have in common that each data point is uniquely assigned to its closest prototype. Therefore, they are also called crisp vector quantizers. Yet, in practical applications, data are often overlapping making it hard to separate clusters. For this kind of data fuzzy vector quantizing, algorithms have been developed, for example, fuzzy c-means (FCM) [4] and fuzzy SOM (FSOM) [5]. Now, each datapoint can be partially assigned to each prototype. The FSOM is an extension of the FCM taking the neighborhood cooperativeness into account. Yet, as common to SOM, this neighborhood is bound to an external topological structure like a grid. In this paper we combined FCM with NG, thus exploiting the advantages of each: fuzziness from FCM and dynamic neighborhood cooperativeness without structural restrictions from NG. Our new approach is called Fuzzy Neural Gas (FNG).

Beside its basic functionality we also introduce some variations of FNG. First, we propose the kernel fuzzy neural gas (KFNG) where we consider differentiable kernels to adapt the metric. This allows the algorithm to operate in the same structural space as support vector machines (SVM) [6], which are known to deliver respectable results [7]. In [6], it has been shown that this modified optimization space is equivalent and isometric to a reproducing kernel Hilbert or Banach space, which proves to be beneficial for unsupervised VQ, that is also for FNG.

For another variant of FNG we were inspired by simulated annealing (SA), a method which allows temporary deterioration of an optimization process to stabilize its long term behavior. To obtain an SA-like approach, we introduce *negative learning* and call the new method pulsing Neural Gas (PNG). The idea can also be transferred to FNG resulting in Pulsing Fuzzy Neural Gas (PFNG).

Clustering in general is an ill-posed problem and it is difficult to validate a cluster solution. Specification the validation of very large data sets, where a cluster might be represented by more than one prototype, turns out to be a challenge. There exist a number of validity measures based on separation and compactness, yet most of them presume that each cluster should be represented by exactly one prototype. Taşdemir and Merényi proposed the ConnIndex [8], which is suited to evaluate crisp clusterings, where each cluster contains more than one prototype. This ConnIndex takes the neighborhood structure between the learned prototypes into account to transfer the information of the full data set to the cluster validation process. We propose a modification for fuzzy cluster solutions and use this Fuzzy ConnIndex in the experimental section.

In the experimental section, we use three different data sets, an artificial one and two real world problems, to compare the cluster solutions obtained by FNG with those obtained by FCM. For evaluation purposes the Fuzzy ConnIndex is applied. Further, we demonstrate the performance of Pulsing Neural Gas on a checkerboard data set. This type of problem is highly multimodal and usually the algorithms do not find all clusters.

2. Fuzzy Neural Gas

The Fuzzy Neural Gas algorithm is a vector quantizer suitable for overlapping data resulting in fuzzy cluster solutions. It is a combination of the Neural Gas (NG) algorithm which incorporates neighborhood relations between data points and prototypes and the Fuzzy c-Means (FCM) which provides a way to obtain fuzzy data point assignments. In the following section, the NG and the FCM are presented shortly to reproduce the derivation of the FNG originally published in [9]. Besides providing an understanding for the principle functioning of the FNG, the description of the basic algorithms is also useful in preparation of Section 4, where a fuzzy cluster validation method called Fuzzy ConnIndex (fConn) is presented.

2.1. Neural Gas. The Neural Gas vector quantizer [3] is an approach which utilizes the dynamic neighborhood between the prototypes $W = \{\mathbf{w}_j\}_{j=1}^{N_P}$, $\mathbf{w}_j \in \mathbb{R}^d$, to obtain a clustering of data samples $\mathbf{v}_i \in \mathbb{R}^d$, $i = 1, \ldots, N_V$, from a data set V. This neighborhood function is based on a winner ranking of the prototypes for each data point. The rank of prototype $\mathbf{w}_j$ is obtained by

$$rk_j\left(\mathbf{v}_i, W\right) = \sum_{l=1}^{N_P} \Theta\left(d\left(\mathbf{v}_i, \mathbf{w}_j\right) - d\left(\mathbf{v}_i, \mathbf{w}_l\right)\right), \tag{1}$$

with the heaviside function $\Theta(x) = 0$, if and only if $x \leq 0$ and 1 else, and a dissimilarity measure $d(\mathbf{v}_i, \mathbf{w}_j)$ which determines the distance between data point $\mathbf{v}_i$ and prototype $\mathbf{w}_j$. Usually the Euclidean distance is used for $d(\mathbf{v}_i, \mathbf{w}_j)$.

The neighborhood $h_\sigma^{\mathrm{NG}}(\mathbf{v}_i, \mathbf{w}_j, W)$ of a data point $\mathbf{v}_i$ is specified by

$$h_\sigma^{\mathrm{NG}}\left(\mathbf{v}_i, \mathbf{w}_j, W\right) = c_\sigma^{\mathrm{NG}} \cdot \exp\left(-\frac{\left(rk_j\left(\mathbf{v}_i, W\right)\right)^2}{2\sigma^2}\right), \tag{2}$$

where the rank rk_j of prototype $\mathbf{w}_j$ is an essential part. For the neighborhood only the prototypes within a certain range σ according to their rank are considered, giving the closest prototype the highest emphasis. The constant c_σ^{NG} is arbitrarily chosen.

The neighborhood can be used to calculate the local costs:

$$lc^{\mathrm{NG}}\left(\mathbf{v}_i, \mathbf{w}_j, W\right) = h_\sigma^{\mathrm{NG}}\left(\mathbf{w}_j, \mathbf{v}_i\right) \cdot d\left(\mathbf{v}_j, \mathbf{w}_j\right), \tag{3}$$

which resemble the local distortions around prototype j weighted by the neighborhood cooperativeness.

The Neural Gas cost function which has to be minimized for optimal clustering directly embeds the local costs:

$$E_{\mathrm{NG}} = \frac{1}{2K(\sigma)} \sum_{j=1}^{N_P} \int P\left(\mathbf{v}_i\right) \underbrace{h_\sigma^{\mathrm{NG}}\left(\mathbf{v}_i, \mathbf{w}_j, W\right) \cdot d(\mathbf{v}_i, \mathbf{w}_j)}_{lc^{\mathrm{NG}}} d\mathbf{v}_i. \tag{4}$$

The normalisation constant $K(\sigma)$ depends on c_σ^{NG} and $P(\mathbf{v}_i)$ is the data density.

The minimization of the cost function (4) is performed by stochastic gradient descent with respect to the prototypes. Given a data point $\mathbf{v}_i$ the prototype update rule yields

$$\Delta\mathbf{w}_j = -\varepsilon \cdot h_\sigma\left(\mathbf{v}_i, \mathbf{w}_j, W\right) \cdot \frac{\partial d\left(\mathbf{v}_i, \mathbf{w}_j\right)}{\partial \mathbf{w}_j}, \tag{5}$$

where $\varepsilon > 0$ is the learning rate [3].

After convergence of the algorithm the whole data set V is approximated by the set W of prototypes. The receptive field Ω_j of each prototype $\mathbf{w}_j$ is defined as

$$\Omega_j = \left\{\mathbf{v}_i \mid d\left(\mathbf{v}_i, \mathbf{w}_j\right) < d\left(\mathbf{v}_i, \mathbf{w}_k\right),\ \forall k\right\}. \tag{6}$$

For crisp clusterings it has been shown in [3] that the NG algorithm results in better cluster solutions than Self-Organizing Maps (SOM) [2] due to its flexible neighborhood compared to the fixed grid of a SOM.

2.2. Fuzzy c-Means. The Fuzzy c-Means [10] is also a vector quantizer where each cluster Ω_j is represented by a prototype $\mathbf{w}_j$ located in its center of gravity. Yet contrary to NG, a data point can be assigned to more than one prototype. The cost function to minimize is given by

$$E_{\mathrm{FCM}} = \sum_{i=1}^{N_V} \sum_{j=1}^{N_P} u_j(\mathbf{v}_i)^m d\left(\mathbf{v}_i, \mathbf{w}_j\right), \tag{7}$$

where the fuzzy assignment of data point $\mathbf{v}_i$ to prototype $\mathbf{w}_j$ is described by $u_j(\mathbf{v}_i) \geq 0$. If the restriction $\sum_{j=1}^{N_P} u_j(\mathbf{v}_i) = 1$ is valid, the clustering is called probabilistic, otherwise possibilistic. The exponent $m > 1$ regulates the fuzziness and according to [10] it should be set to $1.2 \leq m \leq 2$. Again, the distance $d(\mathbf{v}_i, \mathbf{w}_j)$ is usually chosen to be the Euclidean distance.

The algorithm itself is an alternating optimization of prototypes and fuzzy assignments. The update of the prototypes is carried out by keeping the assignments fixed and vice versa the assignments are adapted based on fixed prototypes:

$$\mathbf{w}_j = \frac{\sum_{i=1}^{N_V} u_j(\mathbf{v}_i)^m \cdot \left(\partial d\left(\mathbf{v}_i, \mathbf{w}_j\right) / \partial \mathbf{w}_j\right)}{\sum_{i=1}^{N_V} u_j(\mathbf{v}_i)^m}, \tag{8}$$

$$u_j\left(\mathbf{v}_i\right) = \frac{1}{\sum_{k=1}^{N_P} \left(d(\mathbf{v}_i, \mathbf{w}_j)/(d(\mathbf{v}_i, \mathbf{w}_k)\right)^{1/(m-1)}}. \tag{9}$$

Since the definition of the receptive field (6) does not reflect the information contained in the fuzzy assignments, we define the fuzzy receptive field as

$$\Omega_k^F = \left\{\mathbf{v}_i \mid u_k\left(\mathbf{v}_i\right) > 0\right\}. \tag{10}$$

2.3. Combining NG and FCM to the Fuzzy Neural Gas. As mentioned above the Fuzzy Neural Gas can now be obtained by combining NG and FNG. Thereby, the FCM distance function in (7) is replaced by local costs similar to the NG local costs (3):

$$lc_\sigma^{\mathrm{FNG}}\left(\mathbf{v}_i,\mathbf{w}_j\right)=\sum_{l=1}^{N_P}h_\sigma^{\mathrm{FNG}}\left(\mathbf{w}_j,\mathbf{w}_l\right)\cdot d\left(\mathbf{v}_i,\mathbf{w}_j\right)^2, \tag{11}$$

yielding the cost function

$$E_{\mathrm{FNG}}=\sum_{i=1}^{N_V}\sum_{j=1}^{N_P}u_j(\mathbf{v}_i)^m\underbrace{\sum_{l=1}^{N_P}h_\sigma^{\mathrm{FNG}}\left(\mathbf{w}_j,\mathbf{w}_l\right)\cdot d\left(\mathbf{v}_i,\mathbf{w}_j\right)^2}_{lc_\sigma^{\mathrm{FNG}}(\mathbf{v}_i,\mathbf{w}_j)}. \tag{12}$$

The local costs (11) take the dynamic neighborhood structure according to

$$h_\sigma^{\mathrm{FNG}}\left(\mathbf{w}_j,\mathbf{w}_l\right)=c_\sigma\cdot\exp\left(-\frac{\left(rk_j\left(\mathbf{w}_l,W\right)\right)^2}{2\sigma^2}\right) \tag{13}$$

into account, where the value $\sigma > 0$ is the neighborhood range and c_σ assures that $\sum_l h_\sigma^{\mathrm{FNG}}(\mathbf{w}_j,\mathbf{w}_l)=1$. For optimal performance σ should be decreased adiabatically in the course of optimization. Note that the neighborhood contrary to the NG neighborhood is based on the winning ranks according to the *best matching prototype* and not as known from NG according to the data. The ranks are calculated similar to (1):

$$rk_j\left(\mathbf{w}_l,W\right)=\sum_{k=1}^{N_P}\Theta\left(d\left(\mathbf{w}_l,\mathbf{w}_j\right)-d\left(\mathbf{w}_l,\mathbf{w}_k\right)\right), \tag{14}$$

where $\Theta(x)$ again is the heaviside function.

Analogous to FCM, the update of the prototypes and the fuzzy assignments follows an alternating optimization scheme to minimize the FNG cost function (12). The update scheme consists of two update steps: updating the prototypes while keeping the fuzzy assignments fixed and updating the assignments while retaining the prototypes. The update rules are obtained by Lagrange optimization taking the side condition $\sum_{j=1}^{N_P}u_j(\mathbf{v}_i)=1$ into account.

A batch update considering all the data samples at once is possible if the Euclidean distance is used for the calculation of the local costs (11). The resulting equations can be solved for $\mathbf{w}_j$ and $u_j(\mathbf{v}_i)$, respectively, yielding

$$\begin{aligned}\mathbf{w}_j&=\frac{\sum_{i=1}^{N_V}\sum_{l=1}^{N_P}u_l(\mathbf{v}_i)^m\cdot h_\sigma^{\mathrm{FNG}}\left(\mathbf{w}_j,\mathbf{w}_l\right)\cdot\mathbf{v}_i}{\sum_{i=1}^{N_V}\sum_{l=1}^{N_P}u_l(\mathbf{v}_i)^m\cdot h_\sigma^{\mathrm{FNG}}\left(\mathbf{w}_j,\mathbf{w}_l\right)},\\ u_j\left(\mathbf{v}_i\right)&=\frac{1}{\sum_{l=1}^{N_P}\left(lc_\sigma^{\mathrm{FNG}}\left(\mathbf{v}_i,\mathbf{w}_j\right)/lc_\sigma^{\mathrm{FNG}}\left(\mathbf{v}_i,\mathbf{w}_l\right)\right)^{1/(m-1)}}.\end{aligned} \tag{15}$$

Note that the update of the fuzzy assignments is similar to the FCM assignment update (9) yet instead of the distances $d(\mathbf{v}_i,\mathbf{w}_j)$ the local costs (11) are considered.

For other distances besides the Euclidean distance, the equation obtained by Lagrange optimization might not be solvable for $\mathbf{w}_j$. In that case, the prototypes have to be adapted online via stochastic gradient descent in order to minimize the FNG cost function (12). The corresponding update rule is

$$\Delta\mathbf{w}_j=\sum_{l=1}^{N_P}\sum_{i=1}^{N_V}u_l(\mathbf{v}_i)^m h_\sigma^{\mathrm{FNG}}\left(\mathbf{w}_j,\mathbf{w}_l\right)\frac{\partial d\left(\mathbf{v}_i,\mathbf{w}_j\right)}{\partial\mathbf{w}_j}. \tag{16}$$

Since the derivative of the distance $\partial d(\mathbf{v}_i,\mathbf{w}_j)/\partial\mathbf{w}_j$ has to be considered, the distance measure is required to be differentiable with respect to $\mathbf{w}_j$. Any measure fulfilling this restriction is a suitable measure; that is, alternative to the commonly used Euclidean distance generalized divergences as well as (differentiable) kernels might be used depending on the specific problem at hand. The latter aspect concerning (differentiable) kernels is investigated in detail in the next subsection.

2.4. Fuzzy Neural Gas with Differentiable Kernels. For vector quantizers the distance between prototypes and data samples is determined by a distance measure $d(\mathbf{v}_i,\mathbf{w}_j)$. For FNG this distance has to be differentiable, since the derivative of the distance function $\partial d(\mathbf{v}_i,\mathbf{w}_j)/\partial\mathbf{w}_j$ is considered in the prototype update rule (16) to minimize the cost function. This implies that basically any differentiable distance measure is applicable. The common Euclidean distance can be used as well as generalized divergences [11] or (differentiable) kernels [12]. Each reproducing kernel uniquely corresponds to a kernel feature map $\Phi : V \rightarrow H$, where H is a Hilbert space in a canonical manner [13]. Denote $H' = \Phi(V)$ to be the image of V. The inner product of H is consistent with a kernel; that is, $\kappa_\Phi(\mathbf{v}_i,\mathbf{w}_j)=\langle\Phi(\mathbf{v}_i),\Phi(\mathbf{w}_j)\rangle_H$. Universal continuous kernels ensure the injectivity and continuity of the map. Further, in that case H' is a subspace of H [13]. The inner product defines a metric by

$$\begin{aligned}&d_H\left(\Phi\left(\mathbf{v}_i\right),\Phi\left(\mathbf{w}_j\right)\right)\\&\quad=\sqrt{\kappa_\Phi\left(\mathbf{v}_i,\mathbf{v}_i\right)-2\kappa_\Phi\left(\mathbf{v}_i,\mathbf{w}_j\right)+\kappa_\Phi\left(\mathbf{w}_j,\mathbf{w}_j\right)}.\end{aligned} \tag{17}$$

The nonlinear mapping into the Hilbert space provides large topological richness for the mapped data, which is used for classification in SVMs. However, this topological structure of the image H' may result in better clustering abilities for unsupervised vector quantization.

An example of a universal kernel is the widely known Gaussian kernel:

$$\kappa_\Phi\left(\mathbf{v}_i,\mathbf{w}_j\right)=\exp^{(-\|\mathbf{v}_i-\mathbf{w}_j\|^2/\sigma_g^2)}, \tag{18}$$

where $\|\cdot\|$ is the Euclidean norm. This kernel and the distance metric based thereon can be differentiated easily and is therefore suitable to be used with FNG. A disadvantage is that the parameter σ_g has to be estimated, which is known to be a crucial task.

Another simple yet effective kernel is the ELM kernel (extreme learning machine) [14]. The kernel function is defined as

$$\kappa_{\Phi}\left(\mathbf{v}_i, \mathbf{w}_j\right) = \frac{1}{p}\left\langle \Phi\left(\mathbf{v}_i\right), \Phi\left(\mathbf{w}_j\right)\right\rangle \tag{19}$$

and is simply the normalized dot product in the feature space. In the context of FNG the number p of hidden variables corresponds to the number $Z(H')$ of intrinsic dimensions [15] of H' with $Z(H') \leq N$. In case that the mapping $\Phi(x)$ is not known, for $p \rightarrow \infty$ the kernel can be estimated by an analytic expression [16]:

$$\begin{aligned} &\kappa_{\Phi}\left(\mathbf{v}_i, \mathbf{w}_j\right) \\ &= \frac{2}{\pi}\arcsin\frac{1+\left\langle \mathbf{v}_i, \mathbf{w}_j\right\rangle}{\sqrt{\left(1/2\sigma^2+1+\left\langle \mathbf{v}_i, \mathbf{v}_i\right\rangle\right)\left(1/2\sigma^2+1+\left\langle \mathbf{w}_j, \mathbf{w}_j\right\rangle\right)}}, \end{aligned} \tag{20}$$

which is the so-called asymptotic ELM kernel, where σ is the Gaussian distribution of the data.

3. Pulsing Neural Gas

It has been shown that the Neural Gas algorithm converges to a global minimum in infinite time [3]. Yet in practice, time is limited and prototypes might only have reached a local minimum by the time the algorithm stops.

The proposed method in this section called Pulsing Neural Gas is a combination of NG and Simulated Annealing (SA), another widely known technique for solving optimization problems. SA is a probabilistic metaheuristics which accepts a random solution with a certain probability following the Boltzmann-Gibbs distribution $p(\Delta, T) \sim \exp(-\Delta/T)$. This probability p depends on the difference Δ between a random solution and the former accepted solution and a temperature T which is decreasing over time and convergese to zero. Caused by the cooling respective annealing of the temperature T, towards the end of the optimization process a deterioration of the cost function is accepted with lower probability than at the beginning. This leads to a stable behavior in the periphery of the global minimum.

To transfer this idea to (Fuzzy) Neural Gas a correspondent to the deterioration in SA has to be found. For the common NG the cost function (4) is minimized by performing stochastic gradient descent learning. Although it cannot be guaranteed, on average the value of the cost function decreases which we consider as *positive learning*. We now introduce *negative learning*; that is, we allow the algorithm to perform a *negative learning step*, which increases the cost function temporarily. Hence, on average, the algorithm performs *positive learning*, but once in a while with a certain decreasing probability following a Gibbs-distribution a *negative learning step* causes a disturbance. Possibly this helps to overcome local minima and speeds up convergence to the global minimum.

First considerations took gradient ascent learning into account. However, investigations have shown that this strategy leads to an unstable learning behavior. Instead we suggest a *reverse prototype ranking*:

$$rk_j^-\left(\mathbf{v}_i, W\right) = \left(N_p - 1\right) - \sum_{k=1}^{N_p} H\left(d\left(\mathbf{v}_i, \mathbf{w}_j\right) - d\left(\mathbf{v}_i, \mathbf{w}_k\right)\right), \tag{21}$$

for a given data point $\mathbf{v}_i$. This ranking reverses the known (positive) ranking (1) such that the prototype with the largest distance now becomes the best (lowest) rank (see Figure 1); that is, the update of the prototypes is performed in reverse order and in opposite direction. The prototype update rule is formulated as

$$\Delta\mathbf{w}_j = \varepsilon \cdot h_{\sigma}^- \cdot \frac{\partial d\left(\mathbf{v}_i, \mathbf{w}_j\right)}{\partial \mathbf{w}_j}, \tag{22}$$

where the neighborhood function

$$h_{\sigma}^-\left(j, l\right) = c_{\sigma} \cdot \exp\left(-\frac{\left(rk_j^-\left(\mathbf{w}_l, W\right)\right)^2}{2\sigma^2}\right) \tag{23}$$

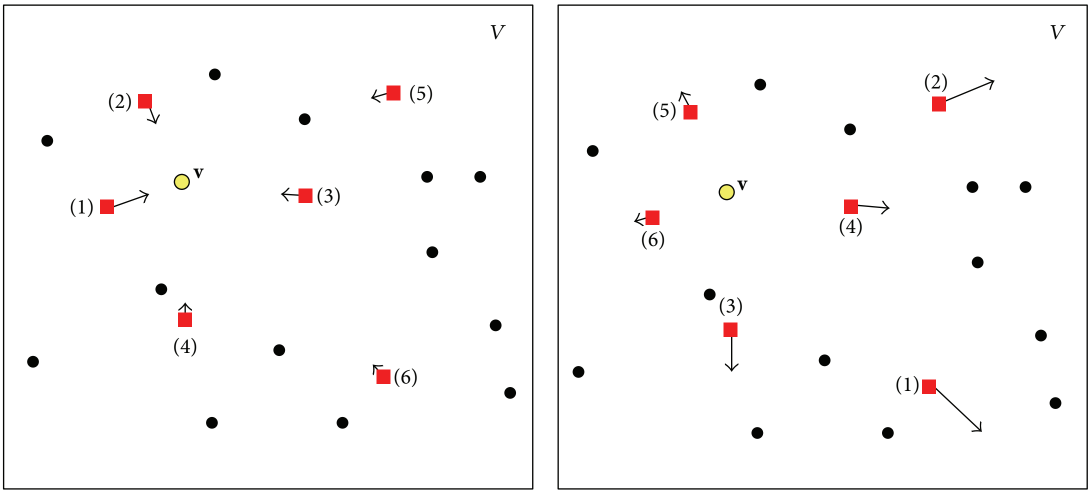
depends on the reverse rankings rk_j^- (21). Now, in contrast to the common positive NG update step, the prototypes are not moved towards the presented data point. Yet instead, according to their reverse ranks they are pushed away, causing little change on the prototypes close to the data point and larger shifts of the prototypes located farther away. Figure 1 depicts this difference between the common NG and the Pulsing NG incorporating negative learning motivated by Simulated Annealing.

Unfortunately, this strategy is not directly transferable to the batch variants of NG [17] and FNG (12). Here all the data points are presented at once and the relocation of the prototypes at each update step depends on all data points. For this variant the idea of Simulated Annealing is performed differently. Instead of a reverse ranking, now only a random subset of the data samples is presented at a randomly chosen update step:

$$\mathbf{w}_j = \frac{\sum_{\mathbf{v}\in A} h_{\lambda}\left(rk_j\left(\mathbf{v}, W\right)\right)\cdot \mathbf{v}}{\sum_{\mathbf{v}\in A} h_{\lambda}\left(rk_j\left(\mathbf{v}, W\right)\right)}, \tag{24}$$

where $A \subset V$ is a nonempty subset. The probability for performing this update step again follows a Gibbs-distribution decreasing with proceeding training. This way, the trend of the relocations is interrupted enabling the prototypes to leave prospective local minima yet possibly causing higher costs temporarily.

One can visualize this procedure as a more or less smooth process approximating some local optimum and once in a while the whole system is shaken up resulting in a temporary increase of the cost function and causing a reorientation of the whole adaptation process. We name this modification of the NG algorithm Pulsing Neural Gas (PNG) and for the fuzzy variant FPNG.



(a)

(b)

Figure 1: Prototype update Neural Gas (a) versus Pulsing Neural Gas (b). NG: the closer the prototype the lower its rank and the stronger the update in direction of the data point. PNG: at random time steps *negative learning* is performed. The closer the prototype the higher its rank and the weaker the update in the opposite direction of the data sample.

4. Fuzzy ConnIndex for the Evaluation of Fuzzy Clusterings

A strategy to cluster very large data sets is to perform vector quantization followed by a clustering of the obtained prototypes. If it can be assured that each of the resulting clusters Ξ_l is represented by more than one prototype, the ConnIndex [8] as proposed by Taşdemir and Merényi can be used for validation purposes. Yet, the ConnIndex is suitable only if crisp vector quantization has been performed in the first step. Since we need a method to evaluate cluster solutions based on fuzzy vector quantization we modified the original ConnIndex. In the following, first we recapitulate the index as proposed by Taşdemir and Merényi and subsequently we derive a fuzzy version of the ConnIndex.

Original ConnIndex. In general, the original ConnIndex balances the overall cluster compactness and separation by combining the intercluster connectivity $C_{inter} \in [0, 1]$ and the intracluster connectivity $C_{intra} \in [0, 1]$:

$$C = C_{intra} \cdot \left(1 - C_{inter}\right). \qquad (25)$$

Thereby, C_{intra} measures the compactness of the clusters and C_{inter} evaluates the separation between them. A value of C close to one suggests a good cluster solution.

For the estimation of the connectivity a nonsymmetric cumulative adjacency matrix $\mathbf{A}$

$$\mathbf{A} = \sum_{i=1}^{N_V} \boldsymbol{\Psi}\left(\mathbf{v}_i\right) \qquad (26)$$

with respect to the receptive fields Ω_j (6) is considered. Here, $\boldsymbol{\Psi}(\mathbf{v}_i)$ is the zero $(N_P \times N_P)$-matrix except the element Ψ_{s_0,s_1} which refers to the best matching unit $s_0(\mathbf{v}_i) = \arg\min_j (d(\mathbf{v}_i, \mathbf{w}_j))$ and the second best matching unit $s_1(\mathbf{v}_i) = \arg\min_{k|k\neq s_0}(d(\mathbf{v}_i, \mathbf{w}_k))$ for data point $\mathbf{v}_i$. The value of this element is set to a positive constant γ_1 usually chosen as $\gamma_1 = 1$. The matrix $\boldsymbol{\Psi}(\mathbf{v}_i)$ is called the response matrix with respect to the data vector $\mathbf{v}_i$. As pointed out in [8], the row vector $\mathbf{a}_j = (a_{j,1}, \ldots, a_{j,N_P})$ of $\mathbf{A}$ describes the density distribution within the receptive field Ω_j with respect to the other $N_P - 1$ prototypes.

The symmetric connectivity matrix

$$\mathbf{C} = \mathbf{A} + \mathbf{A}^{\mathrm{T}} \qquad (27)$$

reflects the topological relations between the prototypes based on the receptive field evaluation. Thereby, the elements $c_{j,k}$ reflect the dissimilarities between the prototypes based on the local data densities.

Now, having the matrices $\mathbf{A}$ and $\mathbf{C}$ defined, the before mentioned connectivities C_{intra} and C_{inter} can be evaluated. The intracluster connectivity C_{intra} is based on the cumulative adjacency matrix $\mathbf{A}$ (26):

$$C_{intra}(l) = \frac{\sum_{j,k|j\neq k}\left\{a_{j,k} \mid \mathbf{w}_j, \mathbf{w}_k \in \Xi_l\right\}}{\sum_{j,k|j\neq k}\left\{a_{j,k} \mid \mathbf{w}_j \in \Xi_l\right\}} \qquad (28)$$

for each cluster Ξ_l. The greater the compactness of a cluster Ξ_l the closer its intraconnectivity is to one. Note again that, as mentioned above, each cluster is made up of more than one prototype $\mathbf{w}_j$.

The inter-cluster connectivity C_{inter} evaluates the separation between the clusters. Analogously, it is the average over the local inter-cluster connectivities

$$C_{inter}(l) = \max_{1\leq m\leq K, m\neq l} C_{inter}(l, m) \qquad (29)$$

of all clusters evaluating the separation of each cluster Ξ_l to the other clusters $\Xi_m, m \neq l$. Thereby, $C_{inter}(l, m)$ judges the separation of cluster Ξ_l to cluster Ξ_m based on the connectivity matrix $\mathbf{C}$ (27) and is defined as

$$C_{inter}(l,m) = \begin{cases} 0 & \text{if } S_{l,m} = 0 \\ \dfrac{\sum_{j,k|j\neq k}\left\{c_{j,k} \mid \mathbf{w}_j \in \Xi_l, \mathbf{w}_k \in \Xi_m\right\}}{\sum_{j,k|j\neq k}\left\{c_{j,k} \mid \mathbf{w}_j \in S_{l,m}\right\}} & \text{if } S_{l,m} \neq 0, \end{cases} \quad (30)$$

where the sets

$$S_{l,m} = \left\{\mathbf{w}_j \mid \mathbf{w}_j \in \Xi_l \wedge \exists \mathbf{w}_k \in \Xi_m : a_{j,k} > 0\right\} \quad (31)$$

describe the neighborhood relations between the clusters Ξ_l and Ξ_m based on the contained prototypes. In contrast to C_{intra}, the value of C_{inter} decreases with better separability.

Generalization of the ConnIndex. The ConnIndex by Taşdemir and Merényi considers the best and second best matching units $s_0(\mathbf{v}_i)$ and $s_1(\mathbf{v}_i)$ only, discarding any information provided by higher ranked prototypes. A generalized version of the index is obtained by incorporating higher winning ranks as known from Neural Gas [3]; see (1). Obviously $rk_{s_0(\mathbf{v}_i)}(\mathbf{v}_i, W) = 0$ is the rank of the best matching prototype. Analogously, the pth winner is denoted by $s_{p-1}(\mathbf{v}_i)$ with rank $rk_{s_p}(\mathbf{v}_i, W) = p$. If it is clear from the context, we will abbreviate $s_p = s_p(\mathbf{v}_i)$ in the following.

To incorporate the higher winning ranks the response matrix $\mathbf{\Psi}(\mathbf{v}_i)$ has to be redefined to involve the *full* response of the whole vector quantizer model for a given input $\mathbf{v}_i$. The new response matrix $\overline{\mathbf{\Psi}}(\mathbf{v}_i)$ is a zero matrix of the same size as $\mathbf{\Psi}(\mathbf{v}_i)$ except the row vector regarding the winner $s_0(\mathbf{v}_i)$. The new response matrix is set to

$$\overline{\mathbf{\Psi}}_{s_0}(\mathbf{v}_i) = \mathbf{r}(\mathbf{v}_i), \quad (32)$$

where $\mathbf{r}(\mathbf{v}_i)$ is the so-called response vector of all prototype responses for a given input $\mathbf{v}_i$. The vector elements $r_j(\mathbf{v}_i)$ of the jth prototype are defined as

$$r_j(\mathbf{v}_i) = \varphi_\omega\left(rk_j(\mathbf{v}_i, W)\right), \quad (33)$$

with $\varphi_\omega(x)$ being an arbitrary monotonically decreasing function in x. A simple choice for this function is the exponential function $\varphi_\omega(x) = \exp(-(1-(id(x))/(n-1))/2\omega^2)$. The parameter ω determines the range of influence and should be determined carefully. If for the vector quantization an algorithm incorporating neighborhood cooperativeness in learning like Neural Gas [3] or self-organizing maps [18] was used, the ω-parameter should be chosen according to the neighborhood range used there. Yet, an alternative approach could be the direct utilization of the distances $d(\mathbf{v}_i, \mathbf{w}_j)$ instead of the winning ranks and $\varphi_\omega(x)$.

This generalized version of the ConnIndex C_g uses for the calculation of the cumulative adjacency matrix $\mathbf{A}$ in (26) the new response matrices $\overline{\mathbf{\Psi}}(\mathbf{v}_i)$ instead of the original response matrices $\mathbf{\Psi}(\mathbf{v}_i)$.

Note that this version is in concordance with the original version, if $\varphi(x)$ is chosen as

$$\varphi(x) = \begin{cases} \gamma_0 & \text{for } x = 0 \\ \gamma_1 & \text{for } x = 1 \\ 0 & \text{else.} \end{cases} \quad (34)$$

Fuzzy ConnIndex. Up to now we assumed that the vector quantization model is based on a crisp mapping. For these models a winner ranking is available and the response information of the network is collected in the response vector $\mathbf{r}(\mathbf{v}_i)$, reflecting the topological relation between the prototypes. In fuzzy vector quantization algorithms this information is no longer available because each data point is gradually assigned to all prototypes. Yet, the fuzzy data point assignments $u_j(\mathbf{v}_i)$ which can be stored in a $N_V \times N_P$ assignment matrix U also reflect the topography of the underlying data. The assignment vector $\mathbf{u}_i$ is then the specific vector of U which contains the assignment value of a data point $\mathbf{v}_i$ to all of the prototypes and is comparable to the response vector $\mathbf{r}(\mathbf{v}_i)$ used for the Generalized ConnIndex C_g. Therefore, the assignments can be used directly to determine the response matrix $\overline{\mathbf{\Psi}}(\mathbf{v}_i)$ by substituting the response vector $\overline{\mathbf{\Psi}}_{s_0}(\mathbf{v}_i)$ in (32). Consequently, the best matching prototype $s_0(\mathbf{v}_i)$ for a given data vector can be seen as the prototype with the highest fuzzy assignment $u_j(\mathbf{v}_i)$:

$$s_0 = \max_j \left\{u_j(\mathbf{v}_i)\right\}. \quad (35)$$

Now, the row vector $\overline{\mathbf{\Psi}}_{s_0}(\mathbf{v}_i)$ of the redefined response matrix $\overline{\mathbf{\Psi}}(\mathbf{v}_i)$ can simply be chosen as the fuzzy response vector $\mathbf{u}_i$:

$$\overline{\mathbf{\Psi}}_{s_0}(\mathbf{v}_i) = \mathbf{u}_i. \quad (36)$$

Again, the cumulative adjacency matrix $\mathbf{A}$ is calculated as before for the original ConnIndex C and the Generalized ConnIndex C_g according to (25). Further calculations remain unaffected.

Hence, the resulting new fuzzy ConnIndex C_f is the counterpart of the generalized ConnIndex C_g in case of fuzzy vector quantization models.

5. Performance

To evaluate the performance of FNG we designed different experiments to compare this method with crisp vector quantizers and the fuzzy c-Means. We also conducted an experiment examining the pulsing FNG. To perform the tests we used artificial and real world data sets.

For the evaluation of the cluster results we used the ConnIndex or the Fuzzy ConnIndex, respectively. This evaluation measure, described in the previous section, is relatively new [8, 19]. But it seems to be well suited for the evaluation of cluster solutions in terms of separation and compactness.

Additionally, for the first *Smiley* dataset we also calculated the Kappa value κ_C [20] which is a measure to judge the agreement of two cluster solutions. A variant thereof κ_F is

Table 1: ConnIndex and κ-values for clusterings of different datasets obtained by crisp and fuzzy vector quantizers. Due to the low number of clusters of the *Smiley* dataset the κ-values can be calculated. There is almost no difference between the crisp methods but a substantial discrepancy between the fuzzy vector quantizer. FNG performs way better than FCM. This observation is supported by evaluating the ConnIndex. To evaluate the cluster solutions of the other two datasets only the ConnIndex was used: for crisp methods the generalized ConnIndex C_G and for fuzzy methods the Fuzzy ConnIndex C_F.

	Smiley		Indian Pine	Colorado
	κ-value	ConnIndex	ConnIndex	ConnIndex
Crisp				
CM	$\kappa_C = 0.978$	$C_G = 0.1478$	$C_G = 0.13$	$C_G = 0.2321$
NG	$\kappa_C = 0.980$	$C_G = 0.2634$	$C_G = 0.21$	$C_G = 0.4207$
Fuzzy				
FCM	$\kappa_F = 0.775$	$C_F = 0.6279$	$C_F = 0.65$	$C_F = 0.6042$
FNG	$\kappa_F = 0.953$	$C_F = 0.9272$	$C_F = 0.72$	$C_F = 0.7228$

suitable for fuzzy data [21]. Unfortunately, this measure can be used only for cluster solutions with a low number of clusters, since the clusters of the different solutions have to be matched, which is hard for clusterings containing a higher number of clusters.

5.1. Artificial Dataset: Smiley. In the first setting we used the *Smiley* data set [19]. This two-dimensional data set consists of three clusters with varying shapes, number of data samples, variances, and distances to each other (see Figure 2). It contains a total of 809 data points.

In the first step we apply c-Means and NG to perform crisp vector quantization and FCM and FNG to perform fuzzy vector quantization with the fuzziness parameter m set to different values $m = \{1.1, \ldots, 2.0\}$. All algorithms result in acceptable solutions. For $m = 1.4$ the FNG cost function settles at the lowest value; FCM reaches the lowest costs for $m = 1.5$. The obtained FNG prototypes are depicted in Figure 2; the FCM results look similar. Visual evaluation confirms an intuitively good distribution in the data space.

A more objective evaluation is obtained with the help of the (Fuzzy) ConnIndex. Yet, to apply this measure the prototypes themselves have to be grouped to clusters of at least two prototypes each. In this simple experiment this step is done manually following the inherent obvious structure of the data set consisting of three clusters.

The obtained ConnValues are listed in Table 1 and show as expected a clear discrepancy between the ConnIndex values obtained by crisp and those obtained by fuzzy vector quantization. This is due to the influence of the data points located in the gaps between the main clusters on the calculation of the index. It is also evident that NG and FNG perform better than c-Means and Fuzzy c-Means, respectively. The overall best ConnIndex value is obtained for FNG, which is less surprising since this algorithm is a combination of FCM and NG, taking beneficial features of each: NG neighborhood and FCM fuzzy assignments.

The Kappa values κ_C [20] and κ_F [21] measure the agreement of two cluster solutions. The closer the values are to one, the higher is the agreement. Comparing the given data structure with the results obtained by the four different clustering methods yields high values indicating substantial to perfect agreement (according to [22]); see Table 1. It can be observed that the two crisp methods NG and c-Means performed almost equally, while the discrepancy between FCM and FNG is remarkable, indicating superior performance of FNG. Note that the values of the crisp and fuzzy solutions cannot be compared to each other since two different κ-measures are applied.

5.2. Practical Example: Indian Pine. Indian Pine is a publicly available data set taken by the NASA Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) consisting of 145 × 145 pixels [23]. Data samples cover 220 bands of 10 nm width from 400 to 2500 nm. Due to the atmospheric water absorption 20 noisy bands can be identified (104–108, 150–163, and 220) and removed safely [24]. The data set is labeled according to 16 identified classes, but we do not use this information for the current experimental setting.

The processing of the data consists of two steps. First vector quantization is performed to position the prototypes. For this step the same algorithms as in the last experiment are used: crisp c-Means and NG and fuzzy FCM and FNG. In the second step the obtained prototypes are grouped by affinity propagation (AP) [25] to be able to apply the (Fuzzy) ConnIndex for evaluation. Special care has to be taken to fulfill the requirement that each cluster (i.e., prototype cluster) is represented by more than one prototype. For this reason a sufficiently high number of prototypes has to be chosen. We set this number to 64 (four times the number of known classes).

The calculation of the Generalized ConnIndex C_G for the crisp methods is straightforward. For the fuzzy variants again the fuzziness parameter m has to be considered carefully. A value of $m = 1.5$ has proven to be favourable. The respective obtained ConnIndex values C_G and C_F are listed in Table 1.

Although the prototype clustering by Affinity Propagation always results in crisp cluster assignments, the clustering based on FNG vector quantization still yields better ConnIndex values than the other methods.

5.3. Practical Example: Colorado. The *Colorado* data set [26] is a LANDSAT TM image from the Colorado area, USA. The image covers a region of about 50 × 50 kilometers

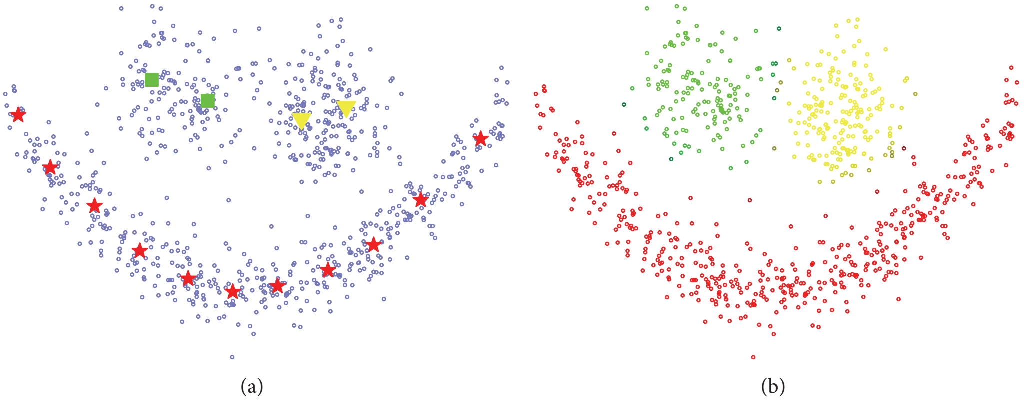


Figure 2: *Smiley* dataset clustered by Fuzzy Neural Gas (FNG). (a) Obtained prototypes. (b) Fuzzy cluster assignments of the data points.

yielding approximately 2 million data points. These are labeled by experts according to 14 different vegetation types and geological formations found in this region. Among them are aspen, mixed pine forest, water, moist meadow, and dry meadow to name a few. The original data samples are 7-dimensional, yet one band (thermal band) is removed due to its low resolution. Generally, the bands are highly correlated [26].

For the experiment we neglected the class information and selected randomly 10% of the data with a representative class distribution. The number of prototypes is set to 56. Besides that, the setup of the experiment is identical to the setup for the *Indian Pine* and consists of the two there described processing steps.

It can be observed that the FCM training is much faster than FNG training and requires less training cycles, about 35 versus 90. Yet, the Fuzzy ConnIndex yields much better results for FNG than FCM (see Table 1) indicating a better prototype distribution in terms of inter- and intraconnectivity of the obtained clusters. The reason for the prolonged processing time can be found in the computational costs to calculate all neighborhood relations anew in each processing step.

5.4. Artificial Data Set: Checkerboard. This artificial data set [27] consisting of compact yet well-separated clusters arranged in a checkerboard-like manner is well suited to demonstrate the performance of the Pulsing Neural Gas compared to the common Neural Gas. The data set contains 11.250 two-dimensional data vectors, which are grouped in 15×15 normally distributed clusters with a standard deviation of $\sigma = 0.3$. The mean distance between two neighboring cluster centers is 2.5. Due to the low dimensionality the data set is well suited for visualization; see Figure 3(a).

For both algorithms NG and PNG all 225 prototypes are initialized in the center of the data set. In the following the algorithms are run both for the same number of steps. For comparison the values of the energy functions according to (4) are used. The experiment showed that on the long run both algorithms performed well. For online learning the effect of the pulsing variant is neglectable, yet the batch version shows significant improvements. The cost function of the Pulsing Neural Gas reaches lower values. The negative learning steps show as little bumps in the plot of the energy functions (see Figure 3(b)), indicating a temporary deterioration. In Figure 3(a) the prototype distribution after 50 learning steps is visualized. Obviously the number of misplaced NG prototypes is higher than the number of misplaced PNG prototypes. This finding is in accordance with the lower value of the PNG energy function.

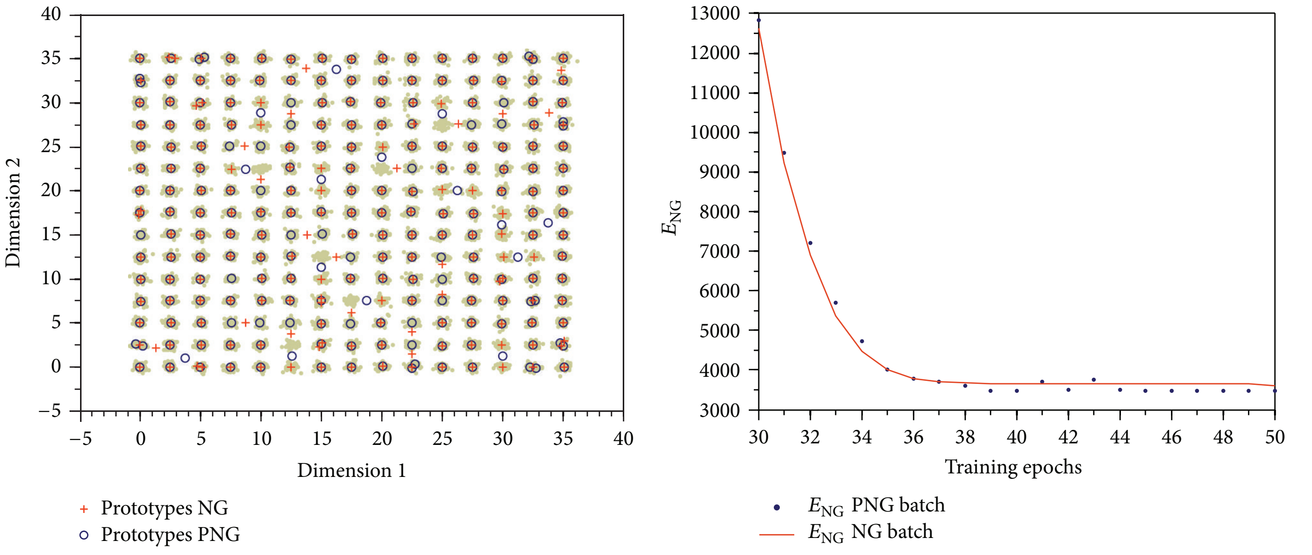
6. Conclusion

We proposed in this paper a fuzzy version of the Neural Gas. By combining the concept of neighborhood cooperativeness as known from NG with the FCM fuzzy assignments we obtain the Fuzzy Neural Gas. This algorithm outperforms FCM by taking dynamic neighborhood relations into account, a paradigm proven to be well suited for crisp vector quantization. The resulting FNG shows good performance compared to standard FCM and crisp NG. Due to the neighborhood cooperativeness this algorithm is insensitive to the initialization of the prototypes.

It is straight forward to introduce other distance measures besides the commonly used Euclidean distance. The only prerequisite is that the measure has to be differentiable; for example, differentiable kernels might be used.

A further variant of NG, respectively, its fuzzy version, is the Pulsing Neural Gas imitating a Simulated Annealing-like behaviour. This modification which allows temporary deterioration of the cost function stabilizes in the long run the learning procedure and helps the algorithm to overcome local minima more easily. This effect was demonstrated on a checkerboard data set, for which it is known that usually the algorithms do not find all clusters.

And finally, we extended the original crisp cluster evaluation ConnIndex [21] to be used for fuzzy clustering. It is based on a generalization of the index considering all prototypes instead of first and second best matching units only. The fuzzy version additionally takes the fuzzy information provided by the fuzzy data point assignments into account. As the original, the Fuzzy ConnIndex requires more than one



(a)

(b)

Figure 3: Final cluster solutions (a) and cost functions (b) for the *Checkerboard* data set using NG and PNG. (a) It can easily be verified that the PNG algorithm has more prototypes placed within the clusters than common NG. The number of training steps for both algorithms was the same. (b) The blue dots refer to the costs for Pulsing Neural Gas. The little bumps at time steps 41 and 43 indicate that a *negative learning* step has occurred.

prototype per cluster. The index was used for the evaluation of the experiments.

Acknowledgment

Marika Kaden was funded by the European Social Fund (ESF), Saxony.

References

[1] G. H. Ball and D. J. Hall, "A clustering technique for summarizing multivariate data," *Behavioral Science*, vol. 12, no. 2, pp. 153–155, 1967.

[2] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[3] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten, "'Neural-gas' network for vector quantization and its application to time-series prediction," *IEEE Transactions on Neural Networks*, vol. 4, no. 4, pp. 558–569, 1993.

[4] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[5] N. B. Karayiannis and J. C. Bezdek, "An integrated approach to fuzzy learning vector quantization and fuzzy c-means clustering," *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 4, pp. 622–628, 1997.

[6] T. Villmann and S. Haase, "A note on gradient based learning in vector quantization usingdifferentiable kernels for hilbert and banach spaces," Machine Learning Report 01/2012, University of Bielefeld, Bielefeld, Germany, 2012.

[7] B. Schölkopf and A. Smola, *Learning with Kernels*, MIT Press, Cambridge, Mass, USA, 2002.

[8] K. Taşdemir and E. Merényi, "A validity index for prototype-based clustering of datasets with complex structures," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 41, no. 4, pp. 1039–1053, 2011.

[9] T. Villmann, T. Geweniger, M. Kästner, M. Lange, and editors, "Fuzzy neural gas for unsupervised vector quantization," in *Artificial Intelligence and Soft Computing*, L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, and J. Zurada, Eds., vol. 7267 of *Lecture Notes in Computer Science*, pp. 350–358, Springer, Heidelberg, Germany, 2012.

[10] J. C. Bezdek, "A convergence theorem for the fuzzy isodata clustering algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2, no. 1, pp. 1–8, 1980.

[11] T. Villmann and S. Haase, "Divergence-based vector quantization," *Neural Computation*, vol. 23, no. 5, pp. 1343–1392, 2011.

[12] T. Villmann, S. Haase, and M. Kstner, "Gradient based learning in vectorquantization using differentiable kernels," in *Advances in Self-Organizing Maps*, P. A. Estévez, J. C. Príncipe, and P. Zegers, Eds., vol. 198 of *Advances InIntelligent Systems and Computing*, pp. 193–204, Springer, Berlin, Germany, 2013.

[13] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, pp. 337–404, 1950.

[14] B. Frénay and M. Verleysen, "Parameter-insensitive kernel in extreme learning for non-linear support vector regression," *Neurocomputing*, vol. 74, no. 16, pp. 2526–2531, 2011.

[15] W. Liebert, *Chaos und Herzdynamik*, Verlag Harri Deutsch, Frankfurt, Germany, 1991.

[16] C. Williams, "Computing with infinite networks," in *Advances in Neural Information Processing Systems*, vol. 9, pp. 295–301, MIT Press, Cambridge, Mass, USA, 1996.

[17] M. Cottrell, S. Ibbou, and P. Letrémy, "SOM-based algorithms for qualitative variables," *Neural Networks*, vol. 17, no. 8-9, pp. 1149–1167, 2004.

[18] T. Kohonen, *Self-Organizing Maps*, vol. 30 of *Information Sciences*, Springer, Berlin, Germany, 1995, (2nd Extended Edition 1997).

[19] T. Geweniger, M. Kästner, M. Lange, and T. Villmann, "Modified conn-index for theevaluation of fuzzy clusterings," in *Proceedings of the European Symposium on Artificial Neural Networks (ESANN '12)*, 2012.

[20] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, pp. 37–46, 1960.

[21] W. Dou, Y. Ren, Q. Wu et al., "Fuzzy kappa for the agreement measure of fuzzy classifications," *Neurocomputing*, vol. 70, no. 4–6, pp. 726–734, 2007.

[22] L. Sachs, *Angewandte Statistikedition*, Springer, New York, NY, USA, 7th edition, 1992.

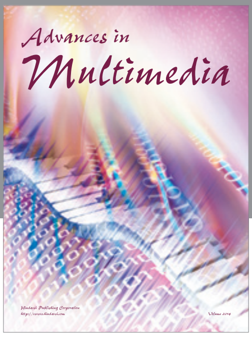
[23] D. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*, Wiley Series in Remote Sensing and Image Processing, John Wiley and Sons, Hoboken, NJ, USA, 2003.

[24] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1351–1362, 2005.

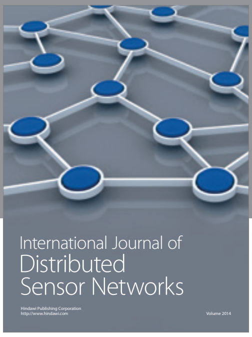
[25] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

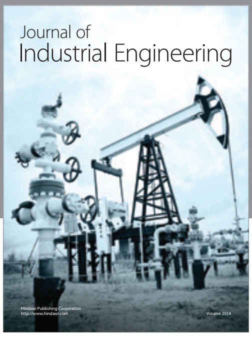
[26] M. F. Augusteijn, K. A. Shaw, and R. J. Watson, "A study of neural network inputdata for ground cover identification in satellite images," in *Proceedings of the International Conference on Artificial Neural Networks (ICANN '93)*, S. Gielen and B. Kappen, Eds., pp. 1010–1013, Springer, London, UK, 1993.

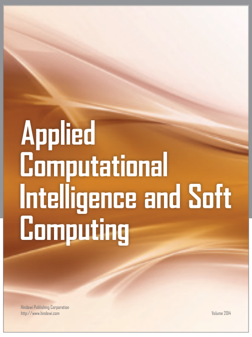
[27] L. Fischer, M. Lange, M. Kästner, and T. Villmann, "Accelerated vector quantization bypulsing neural gas," Machine Learning Reports 6(MLR-04-2012), 2012, http://www.techfak.uni-bielefeld.de/?fschleif/mlr/mlr.

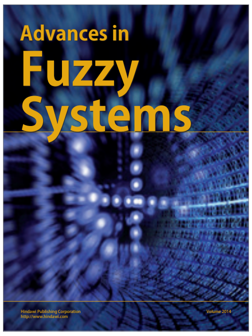

Advances in
Multimedia

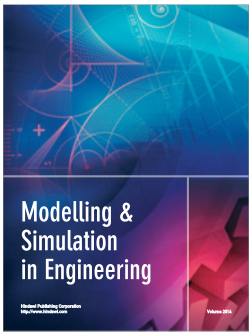
The Scientific
World Journal

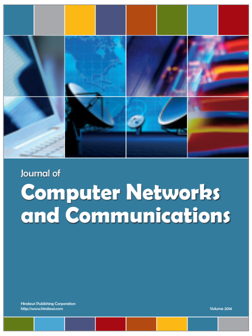

International Journal of
Distributed
Sensor Networks

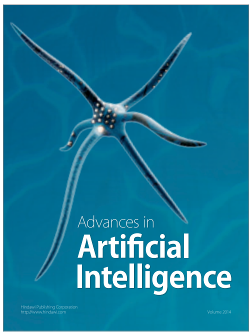

Journal of
Industrial Engineering

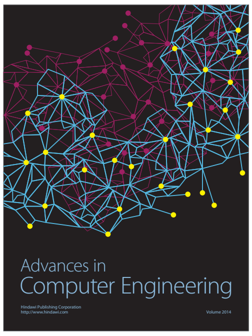

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

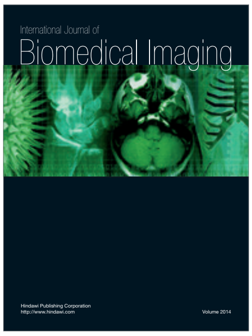

Modelling &
Simulation
in Engineering

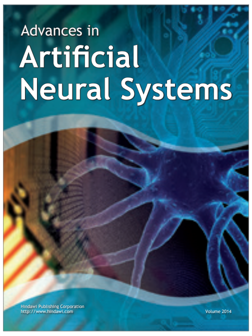

Journal of
Computer Networks
and Communications

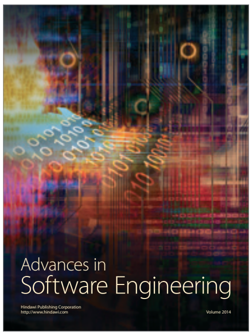

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

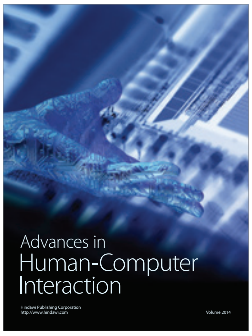
International Journal of
Computer Games
Technology

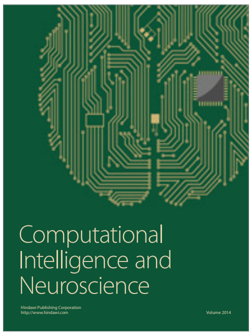

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering