Learning Dynamics and Robustness of Vector Quantization and Neural Gas

Aree Witoelar [a,*] Michael Biehl [a] Anarta Ghosh [b] Barbara Hammer [c]

[a] *University of Groningen, Mathematics and Computing Science, P.O. Box 800, NL-9700 AV Groningen, the Netherlands*
[b] *WaNPRC, University of Washington, Seattle, WA-98195, USA*
[c] *Clausthal University of Technology, Institute of Computer Science, D-98678 Clausthal-Zellerfeld, Germany*

Abstract

Various alternatives have been developed to improve the Winner-Takes-All (WTA) mechanism in vector quantization, including the Neural Gas (NG). However, the behavior of these algorithms including their learning dynamics, robustness with respect to initialization, asymptotic results, etc. has only partially been studied in a rigorous mathematical analysis. The theory of on-line learning allows for an exact mathematical description of the training dynamics in model situations. We demonstrate using a system of three competing prototypes trained from a mixture of Gaussian clusters that the Neural Gas can improve convergence speed and achieves robustness to initial conditions. However, depending on the structure of the data, the Neural Gas does not always obtain the best asymptotic quantization error.

Key words: Vector quantization, Clustering, Online learning, Winner-Takes-All algorithms, Neural Gas

1. Introduction

Vector quantization (VQ) is an important unsupervised learning algorithm, widely used in different areas such as data mining, medical analysis, image compression, and speech or handwriting recognition [1]. The main objective of VQ is to represent the data points by a small number of prototypes or codebook vectors. This can directly be used for compression, clustering, data mining, or (with post-labeling of the prototypes) classification [9,14].

The basic "winner-takes-all" (WTA) or batch algorithms like the popular k-means clustering directly optimize the quantization error underlying vector quantization. However, these methods can be subject to confinement in local minima of the quantization error and can produce suboptimal results. A variety of alternatives to overcome this problem have been proposed, some of which are heuristically motivated while others are based on the minimization of a cost function related to the quantization error: the self-organizing map (SOM) [12], fuzzy-k-means [2], stochastic optimization [7], to name just a few. These algorithms have in common that each pattern influences more than one prototype at a time through a "winner-takes-most" paradigm. Neural gas (NG) as proposed in [13] is a particularly robust variation of vector quantization with the introduction of neighborhood relations. Unlike the self-organizing map, [12], the NG system takes into account the relative distances between prototypes in the input space and not on a predefined lattice.

In practice, NG algorithms yield better solutions than WTA; however, the effect of this strategy on convergence speed or asymptotic behavior has hardly been rigorously investigated so far.

Methods from statistical physics and the theory

* Corresponding author.
Email address: a.w.witoelar@rug.nl (Aree Witoelar).
URL: http://www.cs.rug/∼aree (Aree Witoelar).

of on-line learning [8] allow for an exact mathematical description of learning systems for high dimensional data. In the limit of infinite dimensionality, such systems can be fully described in terms of a few characteristic quantities, the so-called *order parameters*. The evolution of these order parameters along the training procedure is characterized by a set of coupled ordinary differential equations (ODE). By integrating these ODEs, it is possible to analyse the performance of VQ algorithms in terms of stability, sensitivity to initial conditions, and achievable quantization error. This successful approach has also been reviewed in [8,16], among others.

The extension of the theoretical analysis of simple (WTA-based) vector quantization with two prototypes and two clusters introduced in an earlier work [5] is not straightforward. Additional prototypes and clusters introduce more complex interactions in the system that can result in radically different behaviors. Also, the mathematical treatment becomes more involved and requires, for instance, several numerical integrations. Here we introduce an additional prototype and a mixture of clusters. We investigate not only WTA but also the popular Neural Gas approach [13] for vector quantization. This is an important step towards the investigation of general VQ approaches based on neighborhood interaction such as self-organizing maps.

2. Winner-Takes-All and Neural Gas

Assume input data $\xi \in I\!R^N$, generated according to a given probability density function $P(\xi)$. Vector Quantization represents the input data in the same N-dimensional space by a set of prototypes $W = \left\{\mathbf{w}_i \in I\!R^N\right\}_{i=1}^S$. The primary goal of VQ is to find a faithful representation by minimizing the so-called quantization or distortion error

$$E(W) = \frac{1}{2}\int d\xi P(\xi) \sum_{i=1}^{S} d(\xi, \mathbf{w}_i) \prod_{j\neq i} \Theta_{ij} - \frac{1}{2}\int d\xi P(\xi)\xi^2 \qquad (1)$$

where $\Theta_{ij} \equiv \Theta\big(d(\xi, \mathbf{w}_j) - d(\xi, \mathbf{w}_i)\big)$. For each input vector ξ the closest prototype $\mathbf{w}_i$ is singled out by the product of Heaviside functions, $\Theta(x) = 0$ if $x < 0$; 1 else. Here we restrict ourselves to the quadratic Euclidean distance measure $d(\xi, \mathbf{w}_i) = (\xi - \mathbf{w}_i)^2$. The constant $\frac{1}{2}\int d\xi P(\xi)\xi^2$ term is independent of prototype positions and is subtracted for convenience.

The input data is presented sequentially during training and one or more prototypes are updated online. Algorithms studied here can be interpreted as stochastic gradient descent procedures with respect to a cost function $H(W)$ related to $E(W)$. The generalized form reads

$$H(W) = \frac{1}{2}\int d\xi P(\xi) \sum_{i=1}^{S} d(\xi, \mathbf{w}_i) f(r_i) - \frac{1}{2}\int d\xi P(\xi)\xi^2 \qquad (2)$$

where r_i is the rank of prototype $\mathbf{w}_i$ with respect to the distance $d(\xi, \mathbf{w}_i)$, i.e. $r_i = S - \sum_{j\neq i} \Theta_{ij}$. Rank $r_J = 1$ corresponds to the so-called *winner*, i.e. the prototype $\mathbf{w}_J$ closest to the example ξ. The rank function $f(r_i)$ determines the update strength for the set of prototypes and satisfies the normalization $\sum_{i=1}^{S} f(r_i) = 1$; note that it does not depend explicitly on distances but only on the ordering of the prototypes with respect to the current example.

The corresponding stochastic gradient descent in $H(W)$ is of the form

$$\mathbf{w}_i^{\mu} = \mathbf{w}_i^{\mu-1} + \Delta\mathbf{w}_i^{\mu-1} \;\text{ with}$$
$$\Delta\mathbf{w}_i^{\mu-1} = \frac{\eta}{N} f(r_i)(\xi^{\mu} - \mathbf{w}_i^{\mu-1}) \qquad (3)$$

where η is the learning rate and ξ^{μ} is a single example drawn independently at time step μ of the sequential training process. We compare two different algorithms:

(i) **WTA**:
Only one prototype, the winner, is updated for each input. The cost function directly minimizes the quantization error with $H(W) = E(W)$. The corresponding rank function is

$$f_{\mathrm{WTA}}(r_i) = \prod_{j\neq i} \Theta_{ij} \qquad (4)$$

(ii) **Neural Gas**:
The update strength decays exponentially with the rank controlled by a parameter λ. The rank function is $f(r_i) = \frac{1}{C(\lambda)} h_\lambda(r_i)$ where $h_\lambda(r_i) = \exp(-r_i/\lambda)$ and $C(\lambda) = \sum_{r_i=1}^{S} \exp(-r_i/\lambda)$ is a normalization constant. The parameter λ is adjusted during training; it is frequently set large initially and decreased in the course of training. Note that for $\lambda \to 0$ the NG algorithm becomes identical with WTA. We divide $f(r_i)$ according to its ranks as

$$f_{\rm NG}(r_i) = \frac{1}{C(\lambda)} \sum_{k=1}^{S} h_\lambda(k) g_i(k) \tag{5}$$

where $g_i(k) = 1$ if $r_i = k$; 0 else and $\sum_k g_i(k) = 1$. In a model with three prototypes, this can be written in terms of Heaviside functions

$$\begin{aligned} g_i(1) &= \prod_{j\neq i} \Theta_{ij} \\ g_i(2) &= \sum_{k\neq i} \prod_{j\neq k,i} \Theta_{ij}(1-\Theta_{ik}) \\ g_i(3) &= \prod_{j\neq i} \big(1-\Theta_{ij}\big). \end{aligned} \tag{6}$$

3. Model

We choose the model data as a mixture of M spherical Gaussian clusters:

$$\begin{aligned} P(\xi) &= \sum_{\sigma=1}^{M} p_\sigma P(\xi|\sigma) \text{ with} \\ P(\xi|\sigma) &= \frac{1}{(2\pi\upsilon_\sigma)^{N/2}} \exp\left(-\frac{1}{2\upsilon_\sigma}(\xi - \ell_\sigma \mathbf{B}_\sigma)^2\right) \end{aligned} \tag{7}$$

where p_σ are the prior probabilities of each cluster. The components of vector ξ^μ are random numbers according to a Gaussian distribution with mean vectors $\ell_\sigma \mathbf{B}_\sigma$ and variance υ_σ. The mean vectors are orthogonal, i.e. $\mathbf{B}_i \cdot \mathbf{B}_j = \delta_{ij}$ where δ is the Kronecker delta. The parameters ℓ_σ describe the separation between the clusters. This model can be extended for mixtures of many Gaussian clusters with $M < N$ by choosing a coordinate system where the orthonormality conditions $\mathbf{B}_i \cdot \mathbf{B}_j = \delta_{ij}$ are fulfilled.

Note that the Gaussian clusters strongly overlap in high dimensions. The separation between the clusters is apparent only in the $I\!R^M$ subspace spanned by $\{\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_M\}$. It is therefore a non-trivial task to detect the structure of the data in N dimensions.

4. Analysis of Learning Dynamics

We give a brief description of the theoretical framework and refer to [3] for further details. Following the lines of the theory of on-line learning, e.g. [8], in the thermodynamic limit $N \to \infty$ the system can be fully described in terms of a few characteristic quantities, or so-called order parameters. A suitable set of characteristic quantities for the considered learning model is:

$$R_{i\sigma}^\mu = \mathbf{w}_i^\mu \cdot \mathbf{B}_\sigma \;\text{ and }\; Q_{ij}^\mu = \mathbf{w}_i^\mu \cdot \mathbf{w}_j^\mu. \tag{8}$$

Note that $R_{i\sigma}$ are the projections of prototype vectors $\mathbf{w}_i^\mu$ on the center vectors $\mathbf{B}_\sigma$ and Q_{ij}^μ correspond to the self- and cross- overlaps of the prototype vectors.

From the generic update rule defined above, Eq. (3), we can derive the following recursions in terms of the order parameters:

$$\begin{aligned} \frac{R_{i\sigma}^\mu - R_{i\sigma}^{\mu-1}}{1/N} &= \eta f(r_i)\big(b_\sigma^\mu - R_{i\sigma}^{\mu-1}\big) \\ \frac{Q_{ij}^\mu - Q_{ij}^{\mu-1}}{1/N} &= \eta\big(f(r_j)\big(h_i^\mu - Q_{ij}^{\mu-1}\big) + f(r_i)\big(h_j^\mu \\ &\quad - Q_{ij}^{\mu-1}\big)\big) + \eta^2 f(r_i) f(r_j) \frac{(\xi^\mu)^2}{N} \\ &\quad + \mathcal{O}(\frac{1}{N}) \end{aligned} \tag{9}$$

where h_i^μ and b_i^μ are the projections of the input data vector ξ^μ:

$$h_i^\mu = \mathbf{w}_i^{\mu-1} \cdot \xi^\mu \;\text{ and }\; b_\sigma^\mu = \mathbf{B}_\sigma \cdot \xi^\mu. \tag{10}$$

Note that the last two terms in Eq. (9) come from $(\xi^\mu)^2 - h_i^\mu - h_j^\mu + Q_{ij}^{\mu-1}$, where $(\xi^\mu)^2$ is the only term that scales with N.

In the limit $N \to \infty$, the $\mathcal{O}(1/N)$ term can be neglected and the order parameters become *self - averaging* [15] with respect to the random sequence of examples. This means that fluctuations of the order parameters vanish and the system dynamics can be described exactly in terms of their mean values. Also for $N \to \infty$ the rescaled quantity $t \equiv \mu/N$ can be conceived as a continuous time variable. Accordingly, the dynamics can be described by a set of coupled ODE [3,10] after performing an average over the sequence of input data:

$$\begin{aligned} \frac{dR_{i\sigma}}{dt} &= \eta\big(\langle b_\sigma f(r_i)\rangle - \langle f(r_i)\rangle R_{i\sigma}\big) \\ \frac{dQ_{ij}}{dt} &= \eta\big(\langle h_i f(r_j)\rangle - \langle f(r_j)\rangle Q_{ij} + \langle h_j f(r_i)\rangle \\ &\quad - \langle f(r_i)\rangle Q_{ij}\big) + \eta^2 \sum_\sigma \big(p_\sigma \upsilon_\sigma \langle f(r_i) f(r_j)\rangle_\sigma\big) \end{aligned} \tag{11}$$

where $\langle . \rangle$ is the average over the density $P(\xi)$ and the $\langle . \rangle_\sigma$ is the conditional average over $P(\xi|\sigma)$. Here

we exploit the following relation in the last term of $\frac{dQ_{ij}}{dt}$ in Eq. (11):

$$\lim_{N\to\infty}\frac{\langle\xi^2\rangle}{N}=\lim_{N\to\infty}\frac{1}{N}\sum_\sigma p_\sigma(\upsilon_\sigma N+\ell_\sigma^2)=\sum_\sigma p_\sigma\upsilon_\sigma.$$

Exploiting the limit $N\to\infty$ once more, the quantities h_i^μ, b_σ^μ become correlated Gaussian quantities by means of the Central Limit Theorem. Thus the above averages reduce to integrations in $S+M$ dimensions over the joint density $P(\boldsymbol{x})$ where $\boldsymbol{x}=(h_1^\mu,h_2^\mu,\ldots,h_S^\mu,b_1^\mu,b_2^\mu,\ldots,b_M^\mu)$, which are fully specified by first and second moments [3,6]:

$$\begin{aligned}
&\langle h_i^\mu\rangle_\sigma=\ell_\sigma R_{i\sigma}^{\mu-1},\ \langle b_\tau^\mu\rangle_\sigma=\ell_\sigma\delta_{\tau\sigma}\\
&\langle h_i^\mu h_j^\mu\rangle_\sigma-\langle h_i^\mu\rangle_\sigma\langle h_j^\mu\rangle_\sigma=\upsilon_\sigma Q_{ij}^{\mu-1}\\
&\langle b_\tau^\mu b_\rho^\mu\rangle_\sigma-\langle b_\tau^\mu\rangle_\sigma\langle b_\rho^\mu\rangle_\sigma=\upsilon_\sigma\delta_{\tau\rho}\\
&\langle h_i^\mu b_\tau^\mu\rangle_\sigma-\langle h_i^\mu\rangle_\sigma\langle b_\tau^\mu\rangle_\sigma=\upsilon_\sigma R_{i\tau}^{\mu-1}\,.
\end{aligned}\tag{12}$$

Hence the joint density of h_i^μ, b_τ^μ is given in terms of the order parameters defined in Eq. (8). While most of the integrations can be performed analytically, some have to be implemented numerically. See the Appendix for the computations.

Given the averages for a specific rank function $f(r_i)$, cf. Eqs. (B.7) and (B.14) we obtain a closed form expression of ODE. Using the initial conditions $R_{i\sigma}(0), Q_{ij}(0)$, we integrate this system for a given algorithm and get the evolution of order parameters in the course of training, $R_{i\sigma}(t), Q_{ij}(t)$. The behavior of the system depends on the characteristic of the data and the parameters of the learning scheme, i.e. offset of the clusters ℓ_σ, variance within the clusters υ_σ, learning rate η, and for NG, the rank function parameter λ. As shown in [5], this method of analysis is in good agreement with large scale Monte Carlo simulations of the same learning systems for dimensionality as low as $N=200$.

Analogously, the quantization error, Eq. (1), can be expressed in terms of order parameters

$$E(W)=\frac{1}{2}\sum_{i=1}^{S}\Big\langle\prod_{j\neq i}\Theta_{ij}\Big\rangle Q_{ii}-\sum_{i=1}^{S}\Big\langle h_i\prod_{j\neq i}\Theta_{ij}\Big\rangle\tag{13}$$

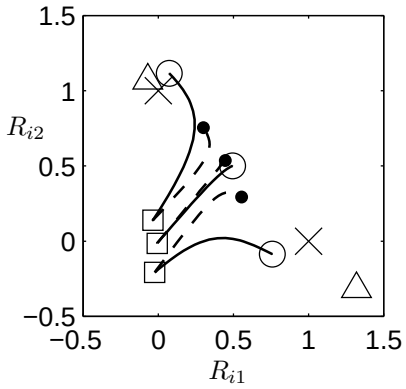
Note that $E(W)$ does not depend explicitly on ξ; here it is shown how the subtracted constant term described in Eq. (1) and Eq. (2) becomes useful.

Plugging in the values of the order parameters computed by integrating the ODEs, $\{R_{i\sigma}(t), Q_{ij}(t)\}$, we can study the so-called learning curve E in dependence of the training time t for a given VQ algorithm.

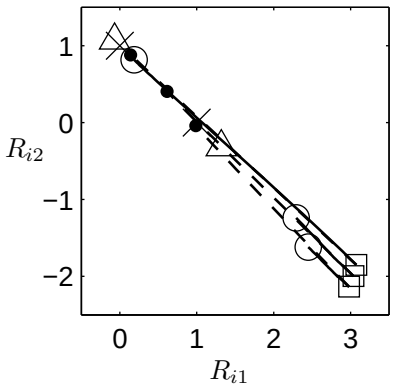
5. Results

5.1. *Learning Dynamics*

We study the performance of both WTA and NG in several cases using three prototypes and up to three clusters. Stochastic gradient descent procedures approach a (local) minimum of the objective function in the limit $\eta\to 0$. We can consider this limit exactly by rescaling the learning time as $\tilde{t}=\eta t$. Then, the $\mathcal{O}(\eta^2)$ terms in Eq. (11) can be neglected and the set of ODEs is simplified. For all demon-

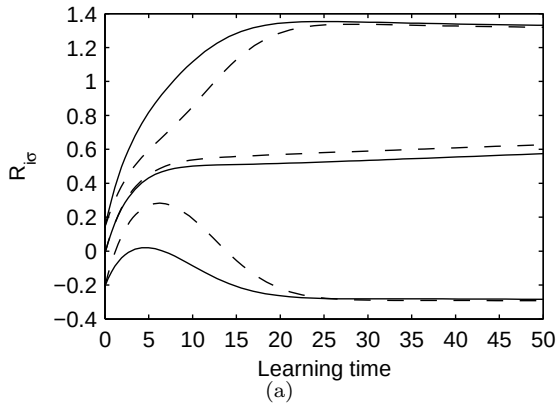


(a)

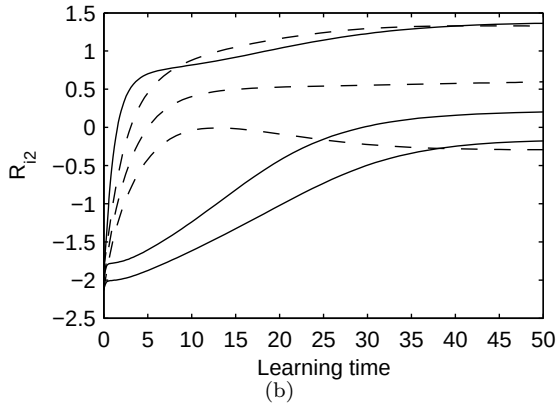


(b)

Fig. 1. Trajectories of prototypes on the plane spanned by $\mathbf{B}_1$ and $\mathbf{B}_2$. The cluster centers $\ell_\sigma\mathbf{B}_\sigma$ are marked by crosses. The trajectories are marked by solid lines (WTA) and dashed lines (NG). The prototypes at initialization are marked with squares and at $\tilde{t}=10$ with circles (WTA) and dots (NG). Both algorithms converge at the triangles, where two prototypes coincide at $\{-0.07, 1.07\}$. The set of prototypes is initially set (a) near the cluster centers, and (b) far away from the cluster centers. In both figures the parameters are $p_1=0.45$, $\ell_1=\ell_2=1$, $\upsilon_1=\upsilon_2=1$, $\eta\to 0$, $\lambda_i=2$, $\lambda_f=0.01$ and $t_f=50$.

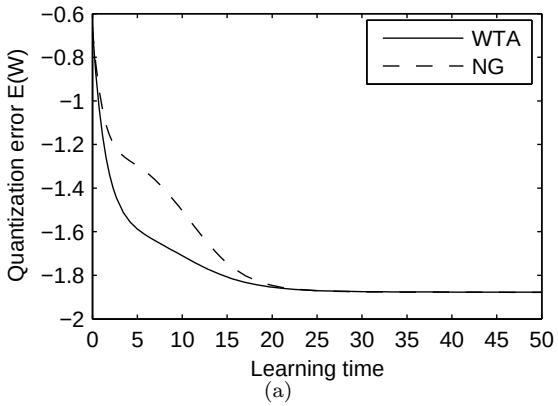


(a)

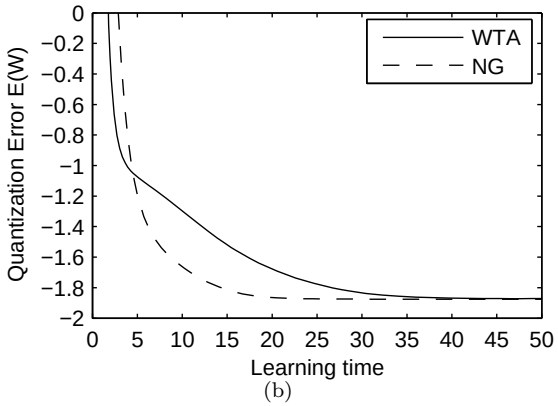


(b)

Fig. 2. The corresponding order parameters R_{i2} at learning time $\tilde{t} = \eta t$ for WTA (solid lines) and NG (dashed lines) algorithms in the system described in Fig.1. The initial sets of prototypes are defined in Figs.1(a) and (b), respectively.



(a)



(b)

Fig. 3. Evolution of the quantization error $E(W)$ in Fig. 1 at learning time $\tilde{t} = \eta t$ for WTA (solid line) and NG (dashed line) algorithms. The prototypes are initialized (a) near the cluster centers and (b) far away from the cluster centers.

strations, the NG algorithm is studied for decreasing λ with $\lambda(\tilde{t}) = \lambda_i(\lambda_f/\lambda_i)^{\tilde{t}/\tilde{t}_f}$ where $\tilde{t}_f$ is a learning time parameter. The influence of the initial set of prototypes on the learning curves is investigated by choosing different values of $\{R_{i\sigma}(0), Q_{ij}(0)\}$.

Figure 1 presents the prototype dynamics in a system with three prototypes and two clusters. We examine two different initial sets of prototypes: close to the origin at $\{R_{i1}(0), R_{i2}(0)\} \approx \{0, 0\}, Q_{ij}(0) \approx 0, \forall\{i, j\}$ in Fig. 1(a); and far away from the origin on the side of the weaker cluster, viz. p_1, at $\{R_{i1}(0), R_{i2}(0)\} \approx \{3, -2\}$, $Q_{ij}(0) = R_{i\sigma}(0) \cdot R_{j\sigma}(0), \forall\{i, j\}$ in Fig. 1(b). While the prototypes have different trajectories in WTA and NG algorithms, they converge at the identical configuration at large t and $\lambda \to 0$. Here, the projections of two prototypes converge near the center of the stronger cluster. The advantage of NG is apparent in Fig. 1(b) where all prototypes already reach the area near the cluster centers at an intermediate learning stage $\tilde{t} = 10$.

This can be illustrated with the evolution of the order parameters $R_{i2}(t)$ in Fig. 2. In Fig. 2(a), the order parameters of both algorithms converge relatively fast. In Fig. 2(b), the order parameters of one prototype change rapidly compared to that of other prototypes in WTA algorithm. One prototype dominates as the winner and gets frequent updates towards the cluster centers, while the other prototypes are rarely updated. The NG algorithm partially solves this problem by updating all prototypes at the initial stages of learning.

The quantization error obtained from the order parameters $\{R_{i\sigma}(\tilde{t}), Q_{ij}(\tilde{t})\}$ is displayed in Fig. 3. We observe that the quantization error decreases faster in the WTA algorithm compared to NG methods at the initial stages of the learning. This behavior can be explained by the fact that the $H_{\rm NG}$ differs from $E(W)$ by smoothing terms in particular in early stages of training. We observe that WTA

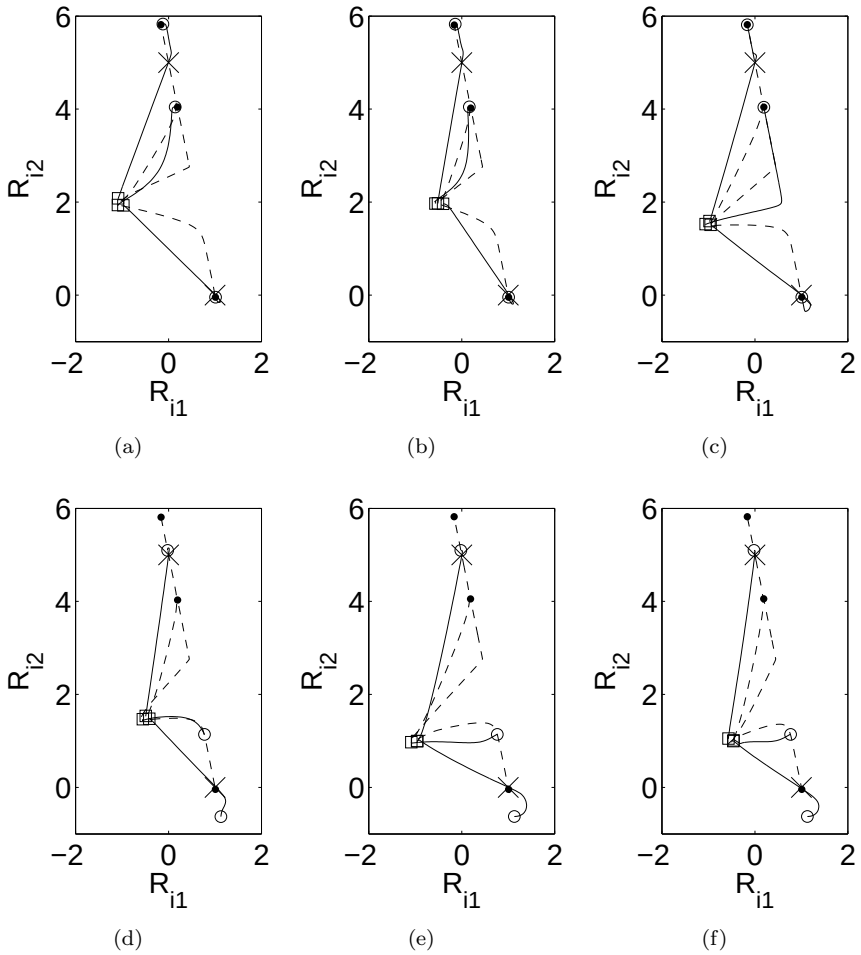


Fig. 4. Trajectories of the prototypes on the plane spanned by $\mathbf{B}_1$ and $\mathbf{B}_2$, corresponding to the WTA (solid lines) and the NG (dashed lines) algorithms. Here, $p_1 = 0.45$, $p_2 = 0.55$, $v_1 = 1$, $v_2 = 1.21$, $\ell_1 = 1$ and $\ell_2 = 5$. The cluster centers $\ell_\sigma \mathbf{B}_\sigma$ are marked by $\times$. The initial prototype configurations for both algorithms are marked with $\square$. While the asymptotic configurations of WTA (circles) algorithm depends on initialization, the NG (dots) always produces identical asymptotic configurations. In these cases, the NG algorithm always finds the optimal quantization error.

yields the best overall quantization error in the first set of initial values in Fig. 3(a). This is mirrored by the fact that, for large $\tilde{t}$ and $\lambda_f \to 0$, both algorithms yield the same quantization error.

For WTA training, the prototypes reach $\tilde{t} \to \infty$ asymptotic positions corresponding to the global minimum of $E(W)$ for small learning rates $\eta \to 0$. However, learning can slow down significantly at intermediate stages of the training process. Transient configurations may persist in the vicinity of local minima and can indeed dominate the training process. The NG is more robust w.r.t. the initial position of prototypes than WTA while achieving the best quantization error asymptotically.

5.2. *Asymptotic configuration*

The dynamics of the prototypes while learning on a model data with a larger separation between the clusters are presented in Fig. 4. The initial configurations correspond to the following values of $\{R_{i1}(0), R_{i2}(0)\}$: (a) $\{-1, 2\}$, (b) $\{-0.5, 2\}$, (c) $\{-1, 1.5\}$, (d) $\{-0.5, 1.5\}$, (e) $\{-1, 1\}$ and (f) $\{-0.5, 1\}$. In all panels, $Q_{ij}(0) = R_{i\sigma}(0) \cdot R_{j\sigma}(0)$.

In this case, the optimal configuration of prototypes is with two prototypes representing the stronger cluster as in Figs. 4(a to c). However, the asymptotic configuration of the prototypes in the WTA algorithm are sensitive to the initial conditions. In some cases, viz. Figs. 4(d to f), this

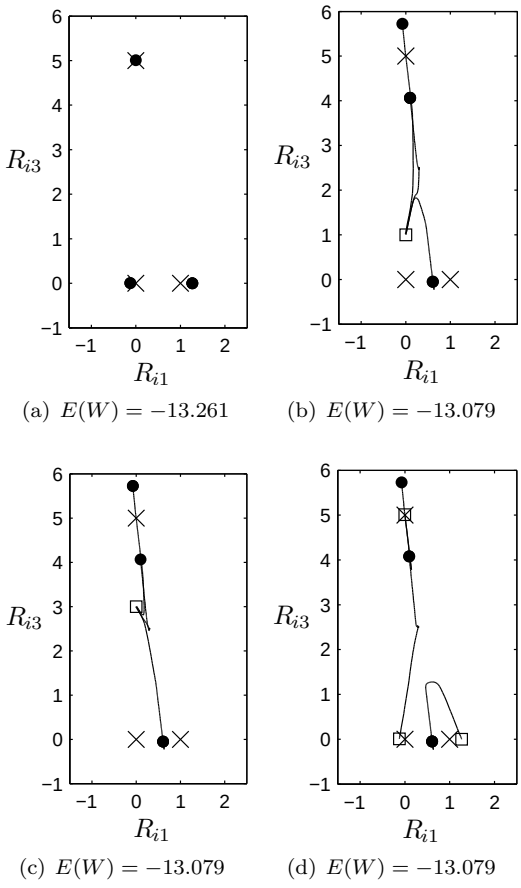
configuration is not the optimal set of prototypes. Therefore, even in this comparably simple model, prototypes in WTA can be confined in suboptimal local minima of the cost function $E(W)$. The issue of different regions of initialization which lead to different asymptotic configurations are to be discussed in forthcoming projects.

The asymptotic configurations for the NG algorithm are independent of initial conditions as shown in Figs. 4(a to f). During the learning process with $\lambda > 0$ the system moves towards intermediate configurations with minimum $H_{\rm NG}(W)$. Given sufficiently large λ and $\tilde{t}$, these configurations are identical and therefore the NG algorithm is robust with respect to initial conditions. In these cases, the asymptotic configuration is the optimal configuration and thus the NG algorithm achieves optimal performance.

We demonstrate a model where the NG algorithm does not yield optimal performance in Fig. 5. In this more complex situation, the weaker cluster ($p_\sigma = 0.45$) is divided into two Gaussian clusters with $p_{1,2} = \{0.25, 0.20\}$. This corresponds to a system of three clusters, with $\ell_\sigma = \{1, 1, 5\}$ and $p_\sigma = \{0.25, 0.20, 0.55\}$. The distance between the first two clusters is small compared to their distance to the third cluster. In comparison to the previous case, where the weaker cluster spreads out evenly in all directions, here it has a particular orientation along the vector $(\mathbf{B}_1 - \mathbf{B}_2)$. Because of this structure, the best quantization error is obtained when one prototype is placed near each cluster center, as in Fig. 5(a), even though one cluster has a very large prior ($p_3 > p_1 + p_2$).

Similar to the previous case, the asymptotic configuration for the NG algorithm is independent of initial conditions. However, this configuration with two prototypes near the center of the stronger cluster in Figs. 5(b to d), is not the optimal configuration. Even with prototypes initialized at the optimal set as in Fig. 5(d), the NG algorithm may still lead to suboptimal configurations.

The characteristics of the cost function $H(W)$ of NG, ie. its minima, can be radically apart with different values of λ. While the NG may find the configuration of the global minima of $H(W)$ for large λ, these configurations do not always lead to the global minima for smaller λ. Consequently, the asymptotic configuration may correspond to a local minimum of $E(W)$ and the NG algorithm does not always yield the optimal quantization error.



(a) $E(W) = -13.261$ (b) $E(W) = -13.079$

(c) $E(W) = -13.079$ (d) $E(W) = -13.079$

Fig. 5. (a) The optimal set of prototypes (solid dots) in a system with three clusters projected on the plane space spanned by $\{\mathbf{B}_1, \mathbf{B}_3\}$. The values of R_{i2} are not shown here. The cluster centers $\ell_\sigma \mathbf{B}_\sigma$ are marked by $\times$. (b,c,d) Trajectories of the prototypes using the NG algorithm with different initial conditions. Their initial (squares) and asymptotic (solid dots) configuration of the prototypes are indicated. The parameters are $p_1 = 0.25, p_2 = 0.20, p_3 = 0.55$, $\upsilon_1 = \upsilon_2 = \upsilon_3 = 1$, $\ell_1 = \ell_2 = 1$ and $\ell_3 = 5$.

6. Conclusion

We have presented an exact mathematical analysis of the dynamics of vector quantization for high dimensional data. Performance is measured by the evolution of the quantization error. In a learning scenario with no sub-optimal local minima of the quantization error, the WTA always converges to the best quantization error. However, learning can slow down significantly if the prototypes are initialized far from the region of high data density. The NG is less sensitive to the initial conditions and achieves both robustness and optimal asymptotic quantization error. Thereby, the convergence speed of NG al-

gorithms is comparable or (for initialization outside the clusters) better than the convergence speed of simple WTA mechanisms, while achieving the same final quantization error.

In the presence of local minima, the WTA algorithm may converge into different asymptotic configurations depending on its initial conditions. The NG algorithm is very robust, i.e. relatively insensitive to initial conditions. However, we demonstrate a test case where it does not find the best asymptotic quantization error. The above discussed suboptimal outcome of NG training might result from the specific schedule at which λ is decreased in the course of training. The influence of both schedules for η and λ will be studied in greater detail in forthcoming projects.

The formalism allows for the design of optimal schemes in the framework of the model situation. While this model clearly does not describe the complexity of real world problems, it is useful to demonstrate certain characteristics of both algorithms. Further extensions could include more realistic data structures, such as additional or non-spherical clusters.

Appendix A. Statistics of the projections

For convenience, we combine the projections h_i and b_σ into an D-dimensional vector where $D = S + M$ as

$$\boldsymbol{x} = \Big(h_1^\mu \;\; h_2^\mu \;\; \ldots \;\; h_S^\mu \;\; b_1^\mu \;\; b_2^\mu \;\; \ldots \;\; b_M^\mu \Big)^T \tag{A.1}$$

The details of the first and second moments are explained in [5] and summarized in Eq. (12). The conditional means $\mu_\sigma = \langle \boldsymbol{x} \rangle_\sigma$ and the conditional covariance matrix $C_\sigma = \langle \boldsymbol{x} \cdot \boldsymbol{x}^T \rangle_\sigma$ can be written in terms of order parameters as

$$\boldsymbol{\mu}_\sigma = \ell_\sigma \;\; \Big(R_{1\sigma} \;\; R_{2\sigma} \;\; \ldots \;\; R_{S\sigma} \;\; \delta_{1\sigma} \;\; \delta_{2\sigma} \;\; \ldots \;\; \delta_{\rho\sigma} \Big)^T \tag{A.2}$$

$$C_\sigma = \upsilon_\sigma \begin{pmatrix} Q_{11} & \cdots & Q_{1S} & R_{11} & \cdots & R_{1\rho} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ Q_{1S} & \cdots & Q_{SS} & R_{S1} & \cdots & R_{S\rho} \\ R_{11} & \cdots & R_{S1} & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ R_{1\rho} & \cdots & R_{S\rho} & 0 & \cdots & 1 \end{pmatrix} \tag{A.3}$$

Appendix B. Averages

Averages of the form $\langle \Theta_{ab} \rangle_\sigma$ in Eq. 11 can be performed analytically, see [3] for details. In contrast to the case of two prototypes only, we encounter additional conditional means of the form

$$\langle \Theta_{ab} \Theta_{cd} \rangle_\sigma \;\text{ and }\; \langle (\mathbf{x})_n \Theta_{ab} \Theta_{cd} \rangle_\sigma$$

where $(\boldsymbol{x})_n$ is the n^{th} component of $\boldsymbol{x}$. The Heaviside functions in Eqs. (4) and (6) are rewritten in the form

$$\begin{aligned} &\Theta_{ab} = \Theta(\boldsymbol{\alpha}_{ab} \cdot \boldsymbol{x} - \beta_{ab}) \;\text{ with} \\ &\boldsymbol{\alpha}_{ab} = (0, \ldots, \underbrace{+2}_{\text{at } a}, \ldots, \underbrace{-2}_{\text{at } b}, \ldots, 0) \\ &\beta_{ab} = Q_{aa} - Q_{bb} \; . \end{aligned} \tag{B.1}$$

The averages are then calculated as follows

$$\begin{aligned} &\langle \Theta_{ab} \Theta_{cd} \rangle_\sigma \\ &= \frac{1}{(2\pi)^{D/2} (\det C_\sigma)^{1/2}} \int_{\mathbb{R}^D} \Theta \left(\boldsymbol{\alpha}_{ab} \cdot \boldsymbol{x} - \beta_{ab} \right) \\ &\quad \times \Theta \left(\boldsymbol{\alpha}_{cd} \cdot \boldsymbol{x} - \beta_{cd} \right) \exp \big(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_\sigma)^T \\ &\quad \times C_\sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_\sigma) \big) d\boldsymbol{x} \\ &= \frac{1}{(2\pi)^{D/2} (\det C_\sigma)^{1/2}} \int_{\mathbb{R}^D} \Theta \big(\boldsymbol{\alpha}_{ab} \cdot \boldsymbol{x'} + \boldsymbol{\alpha}_{ab} \cdot \boldsymbol{\mu}_\sigma \\ &\quad - \beta_{ab} \big) \, \Theta \big(\boldsymbol{\alpha}_{cd} \cdot \boldsymbol{x'} + \boldsymbol{\alpha}_{cd} \cdot \boldsymbol{\mu}_\sigma - \beta_{cd} \big) \\ &\quad \times \exp \big(-\frac{1}{2} \boldsymbol{x'}^T C_\sigma^{-1} \boldsymbol{x'} \big) d\boldsymbol{x'} \\ &\quad \text{(with the substitution } \boldsymbol{x'} = \boldsymbol{x} - \boldsymbol{\mu}_\sigma \text{)}. \end{aligned} \tag{B.2}$$

Because the covariance matrix C_σ is positive definite, $C_\sigma^{1/2}$ exists. Defining $\boldsymbol{x'} = C_\sigma^{1/2} \boldsymbol{y}$, we obtain $\boldsymbol{x'}^T C_\sigma^{-1} \boldsymbol{x'} = \boldsymbol{y}^2$, $d\boldsymbol{x'} = (\det C_\sigma)^{1/2} d\boldsymbol{y}$ and

$$\begin{aligned} &\langle \Theta_{ab} \Theta_{cd} \rangle_\sigma \\ &= \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} \Theta \left(\boldsymbol{\alpha}_{ab} C_\sigma^{1/2} \boldsymbol{y} + \boldsymbol{\alpha}_{ab} \cdot \boldsymbol{\mu}_\sigma - \beta_{ab} \right) \\ &\quad \times \Theta \left(\boldsymbol{\alpha}_{cd} C_\sigma^{1/2} \boldsymbol{y} + \boldsymbol{\alpha}_{cd} \cdot \boldsymbol{\mu}_\sigma - \beta_{cd} \right) \exp(-\frac{1}{2} \boldsymbol{y}^2) d\boldsymbol{y} \\ &= \frac{1}{(2\pi)^{D/2}} \int_{\mathbb{R}^D} \Theta \big(\boldsymbol{\alpha}_{ab} C_\sigma^{1/2} \boldsymbol{y} + \tilde{\beta}_{ab,\sigma} \big) \\ &\quad \times \Theta \big(\boldsymbol{\alpha}_{cd} C_\sigma^{1/2} \boldsymbol{y} + \tilde{\beta}_{cd,\sigma} \big) \exp(-\frac{1}{2} \boldsymbol{y}^2) d\boldsymbol{y} \\ &\quad \text{(where } \tilde{\beta}_{ab,\sigma} = \boldsymbol{\alpha}_{ab} \cdot \boldsymbol{\mu}_\sigma - \beta_{ab} \text{)}. \end{aligned} \tag{B.3}$$

Since $\exp(-\frac{1}{2} \boldsymbol{y}^2)$ has rotational invariance, it is possible to rotate the orthonormal coordinate system

$\boldsymbol{y} = (y_1\boldsymbol{e}_1 + y_2\boldsymbol{e}_2 + \ldots + y_N\boldsymbol{e}_N)$ into $\boldsymbol{y}' = (y'_1\boldsymbol{e}'_1 + y'_2\boldsymbol{e}'_2 + \ldots + y'_N\boldsymbol{e}'_N)$ where one axis, $\boldsymbol{e}'_1$, is aligned with $\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}$ and another axis, $\boldsymbol{e}'_2$, lies on the plane spanned by $\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}$ and $\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}$. This is done by the Gram-Schmidt orthonormal transformation:

$$\begin{aligned}\boldsymbol{e}'_1 &= \frac{\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}}{\|\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\|}\\ \boldsymbol{e}'_2 &= \frac{\boldsymbol{\alpha}_{cd}C_\sigma^{1/2} - (\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\cdot\boldsymbol{e}'_1)\boldsymbol{e}'_1}{\|\boldsymbol{\alpha}_{cd}C_\sigma^{1/2} - (\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\cdot\boldsymbol{e}'_1)\boldsymbol{e}'_1\|}\,. \end{aligned} \tag{B.4}$$

The other axes $\{\boldsymbol{e}'_3, \boldsymbol{e}'_4, \ldots, \boldsymbol{e}'_N\}$ are orthogonal to both $\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}$ and $\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}$ and can be integrated over using the substitution $\frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}}\exp(-\frac{1}{2}z^2)dz = 1$. We obtain from Eq. (B.3),

$$\begin{aligned}\langle\Theta_{ab}\Theta_{cd}\rangle_\sigma = \frac{1}{(2\pi)}\int_{\mathbb{R}^2}&\Theta\left(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\cdot y'_1\boldsymbol{e}'_1 + \tilde{\beta}_{ab,\sigma}\right)\\ &\times\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\cdot(y'_1\boldsymbol{e}'_1 + y'_2\boldsymbol{e}'_2) + \tilde{\beta}_{cd,\sigma}\big)\\ &\times\exp\left(-\frac{1}{2}(y_1'^2 + y_2'^2)\right)dy'_1dy'_2\,.\end{aligned} \tag{B.5}$$

We examine the Heaviside functions $\Theta(x) = 1$ if $x > 0$; 0 else. $\Theta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y'} + \tilde{\beta}_{ab,\sigma}\big) = 1$ and $\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\boldsymbol{y'} + \tilde{\beta}_{cd,\sigma}\big) = 1$ if the following conditions are satisfied

$$\begin{aligned}&y'_1 > y_1^*\ \text{ with }\ y_1^* = -\frac{\tilde{\beta}_{ab,\sigma}}{\tilde{\alpha}_{ab,\sigma}}\\ &y'_2 > y_2^*\ \text{ with }\ y_2^* = \frac{-\tilde{\beta}_{cd,\sigma}\tilde{\alpha}_{ab,\sigma} - \boldsymbol{\alpha}_{ab}C_\sigma\boldsymbol{\alpha}_{cd}\,y'_1}{\sqrt{\tilde{\alpha}^2_{cd,\sigma}\tilde{\alpha}^2_{ab,\sigma} - (\boldsymbol{\alpha}_{cd}C_\sigma\boldsymbol{\alpha}_{ab})^2}}\,.\end{aligned}$$

where we defined $\tilde{\alpha}_{ab,\sigma} = \|\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\|$. Substituting the conditions into Eq. (B.5), we get

$$\begin{aligned}&\langle\Theta_{ab}\Theta_{cd}\rangle_\sigma\\ &= \frac{1}{(2\pi)}\int_{y_1^*}^{\infty}\int_{y_2^*}^{\infty}\exp\left(-\frac{1}{2}\left(y_1'^2 + y_2'^2\right)\right)dy'_2dy'_1\\ &= \frac{1}{(2\pi)}\int_{y_1^*}^{\infty}\exp\left(-\frac{1}{2}y_1'^2\right)\\ &\quad\times\left(\int_{y_2^*}^{\infty}\exp\left(-\frac{1}{2}y_2'^2\right)dy'_2\right)dy'_1.\end{aligned} \tag{B.6}$$

We get the final result in closed form as

$$\begin{aligned}&\langle\Theta_{ab}\Theta_{cd}\rangle_\sigma\\ &= \frac{1}{\sqrt{2\pi}}\int_{-\frac{\tilde{\beta}_{ab,\sigma}}{\tilde{\alpha}_{ab,\sigma}}}^{\infty}\exp\left(-\frac{1}{2}y_1'^2\right)\\ &\quad\times\Phi\left(\frac{\tilde{\beta}_{cd,\sigma}\tilde{\alpha}_{ab,\sigma} + (\boldsymbol{\alpha}_{cd}C_\sigma\boldsymbol{\alpha}_{ab})y'_1}{\sqrt{\tilde{\alpha}^2_{cd,\sigma}\tilde{\alpha}^2_{ab,\sigma} - (\boldsymbol{\alpha}_{cd}C_\sigma\boldsymbol{\alpha}_{ab})^2}}\right)dy'_1\end{aligned} \tag{B.7}$$

with $\Phi(x) = \int_{-\infty}^{x}\frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}a^2)\,da$. The one-fold integration in Eq. (B.7) has to be performed numerically.

The remaining average to be computed is

$$\begin{aligned}&\langle(\mathbf{x})_n\Theta_{ab}\Theta_{cd}\rangle_\sigma\\ &= \frac{1}{(2\pi)^{D/2}(\det C_\sigma)^{1/2}}\int_{\mathbb{R}^D}(\boldsymbol{x})_n\Theta\big(\boldsymbol{\alpha}_{ab}\cdot\boldsymbol{x} - \beta_{ab}\big)\\ &\quad\times\Theta\big(\boldsymbol{\alpha}_{cd}\cdot\boldsymbol{x} - \beta_{cd}\big)\exp\big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_\sigma)^T\\ &\quad\times C_\sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_\sigma)\big)d\boldsymbol{x}\,.\end{aligned} \tag{B.8}$$

Similar to Eq. (B.3), we obtain the form

$$\begin{aligned}&\langle(\mathbf{x})_n\Theta_{ab}\Theta_{cd}\rangle_\sigma\\ &= \frac{1}{(2\pi)^{D/2}}\int_{\mathbb{R}^D}(C_\sigma^{1/2}\boldsymbol{y})_n\Theta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y} + \tilde{\beta}_{ab,\sigma}\big)\\ &\quad\times\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\boldsymbol{y} + \tilde{\beta}_{cd,\sigma}\big)\exp(-\frac{1}{2}\boldsymbol{y}^2)d\boldsymbol{y}\\ &\quad+(\boldsymbol{\mu}_\sigma)_n\langle\Theta_{ab}\Theta_{cd}\rangle_\sigma\\ &= I + (\boldsymbol{\mu}_\sigma)_n\langle\Theta_{ab}\Theta_{cd}\rangle_\sigma\\ &\quad\text{(where } I \text{ is an integral to be computed).}\end{aligned} \tag{B.9}$$

Consider the integrals contributing to I

$$\begin{aligned}I_j = \int_{\mathbb{R}}&(C_\sigma^{1/2})_{nj}(\boldsymbol{y})_j\Theta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y} + \tilde{\beta}_{ab,\sigma}\big)\\ &\times\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\boldsymbol{y} + \tilde{\beta}_{cd,\sigma}\big)\\ &\times\exp(-\frac{1}{2}(\boldsymbol{y})_j^2)d(\boldsymbol{y})_j\end{aligned} \tag{B.10}$$

we perform integration by parts $\int udv = uv - \int vdu$ with

$$\begin{aligned}u &= \Theta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y} + \tilde{\beta}_{ab,\sigma}\big)\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\boldsymbol{y} + \tilde{\beta}_{cd,\sigma}\big)\,,\\ v &= (C_\sigma^{1/2})_{nj}\exp(-\frac{1}{2}(\boldsymbol{y})_j^2)\,,\end{aligned}$$

$$
\begin{aligned}
du =& \frac{\partial}{\partial \boldsymbol{y}_j}\Big(\Theta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{ab,\sigma}\big)\Big)\\
&\times\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{cd,\sigma}\big)d(\boldsymbol{y})_j\\
&+\Theta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{ab,\sigma}\big)\\
&\times\frac{\partial}{\partial \boldsymbol{y}_j}\Big(\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{cd,\sigma}\big)\Big)d(\boldsymbol{y})_j\ ,\\
dv =& -(C_\sigma^{1/2})_{nj}(\boldsymbol{y})_j\exp(-\frac{1}{2}(\boldsymbol{y})_j^2)d(\boldsymbol{y})_j\ ,
\end{aligned}
$$

and we obtain

$$
\begin{aligned}
I_j =&\Big[-\Theta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{ab,\sigma}\big)\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{cd,\sigma}\big)\\
&\times(C_\sigma^{1/2})_{nj}\exp\big(-\frac{1}{2}(\boldsymbol{y})_j^2\big)\Big]_{-\infty}^{\infty}+\int_{\mathbb{R}}(C_\sigma^{1/2})_{nj}\\
&\times\exp(-\frac{1}{2}(\boldsymbol{y})_j^2)\Big[\frac{\partial}{\partial \boldsymbol{y}_j}\Big(\Theta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{ab,\sigma}\big)\Big)\\
&\times\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{cd,\sigma}\big)+\Theta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{ab,\sigma}\big)\\
&\times\frac{\partial}{\partial \boldsymbol{y}_j}\Big(\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{cd,\sigma}\big)\Big)\Big]d(\boldsymbol{y})_j
\end{aligned}
$$

$$
\begin{aligned}
=&(C_\sigma^{1/2})_{nj}\Big(\int_{\mathbb{R}}\frac{\partial\Theta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{ab,\sigma}\big)}{\partial \boldsymbol{y}_j}\\
&\times\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{cd,\sigma}\big)\exp(-\frac{1}{2}(\boldsymbol{y})_j^2)d(\boldsymbol{y})_j\\
&+\int_{\mathbb{R}}\Theta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{ab,\sigma}\big)\\
&\times\frac{\partial\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{cd,\sigma}\big)}{\partial \boldsymbol{y}_j}\exp(-\frac{1}{2}(\boldsymbol{y})_j^2)d(\boldsymbol{y})_j\Big)\ .
\end{aligned}
\tag{B.11}
$$

The sum over j gives

$$
\begin{aligned}
I=&\frac{1}{(2\pi)^{D/2}}\sum_{j=1}^{D}(C_\sigma^{1/2})_{nj}\Big(\int_{\mathbb{R}^D}\frac{\partial\Theta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{ab,\sigma}\big)}{\partial \boldsymbol{y}_j}\\
&\times\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{cd,\sigma}\big)\exp(-\frac{1}{2}\boldsymbol{y}^2)d\boldsymbol{y}\\
&+\int_{\mathbb{R}^D}\Theta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{ab,\sigma}\big)\frac{\partial\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{cd,\sigma}\big)}{\partial \boldsymbol{y}_j}\\
&\times\exp(-\frac{1}{2}\boldsymbol{y}^2)d\boldsymbol{y}\Big)
\end{aligned}
$$

$$
\begin{aligned}
=&\frac{1}{(2\pi)^{D/2}}\Big((C_\sigma\boldsymbol{\alpha}_{ab})_n\int_{\mathbb{R}^D}\delta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{ab,\sigma}\big)\\
&\times\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{cd,\sigma}\big)\exp(-\frac{1}{2}\boldsymbol{y}^2)d\boldsymbol{y}\\
&+(C_\sigma\boldsymbol{\alpha}_{cd})_n\int_{\mathbb{R}^D}\delta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{cd,\sigma}\big)\\
&\times\Theta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{ab,\sigma}\big)\exp(-\frac{1}{2}\boldsymbol{y}^2)d\boldsymbol{y}\Big)\ .
\end{aligned}
\tag{B.12}
$$

where $\delta(\cdot)$ is the Dirac-delta function. In the last step we have used

$$
\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{y}_j}\Big(\Theta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{ab,\sigma}\big)\Big)\\
&=\sum_{i=1}^{D}(\boldsymbol{\alpha}_{ab})_i(C_\sigma^{1/2})_{ij}\Big(\delta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{ab,\sigma}\big)\Big)\ .
\end{aligned}
$$

Calculating the first term only,

$$
\begin{aligned}
I_{ab}=&\frac{1}{(2\pi)^{D/2}}(C_\sigma\boldsymbol{\alpha}_{ab})_n\int_{\mathbb{R}^D}\delta\big(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{ab,\sigma}\big)\\
&\times\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\boldsymbol{y}+\tilde{\beta}_{cd,\sigma}\big)\exp(-\frac{1}{2}\boldsymbol{y}^2)d\boldsymbol{y}\ ,
\end{aligned}
$$

we rotate the coordinate system as in Eq. (B.5) and obtain the following

$$
\begin{aligned}
I_{ab}=&\frac{1}{(2\pi)}(C_\sigma\boldsymbol{\alpha}_{ab})_n\int_{\mathbb{R}^2}\delta\left(\boldsymbol{\alpha}_{ab}C_\sigma^{1/2}\cdot y_1'\boldsymbol{e}_1'+\tilde{\beta}_{ab,\sigma}\right)\\
&\times\Theta\big(\boldsymbol{\alpha}_{cd}C_\sigma^{1/2}\cdot(y_1'\boldsymbol{e}_1'+y_2'\boldsymbol{e}_2')+\tilde{\beta}_{cd,\sigma}\big)\\
&\times\exp\left(-\frac{1}{2}(y_1'^2+y_2'^2)\right)dy_1'dy_2'\\
=&\frac{(C_\sigma\boldsymbol{\alpha}_{ab})_n}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\delta\left(\tilde{\alpha}_{ab,\sigma}y_1'+\tilde{\beta}_{ab,\sigma}\right)\exp\left(-\frac{1}{2}y_1'^2\right)\\
&\times\Phi\left(\frac{\tilde{\beta}_{cd,\sigma}\tilde{\alpha}_{ab,\sigma}+(\boldsymbol{\alpha}_{cd}C_\sigma\boldsymbol{\alpha}_{ab})y_1'}{\sqrt{\tilde{\alpha}_{cd,\sigma}^2\tilde{\alpha}_{ab,\sigma}^2-(\boldsymbol{\alpha}_{cd}C_\sigma\boldsymbol{\alpha}_{ab})^2}}\right)dy_1'\ .
\end{aligned}
$$

Substituting $z=\tilde{\alpha}_{ab,\sigma}y_1'$,

$$
\begin{aligned}
I_{ab}=&\frac{(C_\sigma\boldsymbol{\alpha}_{ab})_n}{\sqrt{(2\pi)}}\int_{-\infty}^{\infty}\delta\left(z+\tilde{\beta}_{ab,\sigma}\right)\\
&\times\exp\left(-\frac{1}{2}\left(\frac{z}{\tilde{\alpha}_{ab,\sigma}}\right)^2\right)\\
&\times\Phi\left(\frac{\tilde{\beta}_{cd,\sigma}\tilde{\alpha}_{ab,\sigma}^2+(\boldsymbol{\alpha}_{cd}C_\sigma\boldsymbol{\alpha}_{ab})z}{\tilde{\alpha}_{ab,\sigma}\sqrt{\tilde{\alpha}_{cd,\sigma}^2\tilde{\alpha}_{ab,\sigma}^2-(\boldsymbol{\alpha}_{cd}C_\sigma\boldsymbol{\alpha}_{ab})^2}}\right)\\
&\times\frac{dz}{\tilde{\boldsymbol{\alpha}}_{ab,\sigma}}
\end{aligned}
$$

$$= \frac{(C_\sigma \boldsymbol{\alpha}_{ab})_n}{\sqrt{(2\pi)}\tilde{\boldsymbol{\alpha}}_{ab,\sigma}} \exp\left(-\frac{1}{2}\frac{\tilde{\beta}^2_{ab,\sigma}}{\tilde{\alpha}^2_{ab,\sigma}}\right) \times \Phi\left(\frac{\tilde{\beta}_{cd,\sigma}\tilde{\alpha}^2_{ab,\sigma} - \tilde{\beta}_{ab,\sigma}(\boldsymbol{\alpha}_{cd} C_\sigma \boldsymbol{\alpha}_{ab})}{\tilde{\alpha}_{ab,\sigma}\sqrt{\tilde{\alpha}^2_{cd,\sigma}\tilde{\alpha}^2_{ab,\sigma} - (\boldsymbol{\alpha}_{cd} C_\sigma \boldsymbol{\alpha}_{ab})^2}}\right) . \quad \text{(B.13)}$$

Analogously we compute the second term in Eq. (B.12) and obtain the final form

$$\begin{aligned}
&\langle (\mathbf{x})_n \Theta_{ab} \Theta_{cd} \rangle_\sigma \\
&= \frac{(C_\sigma \boldsymbol{\alpha}_{ab})_n}{\sqrt{(2\pi)}\tilde{\alpha}_{ab,\sigma}} \exp\left(-\frac{1}{2}\frac{\tilde{\beta}^2_{ab,\sigma}}{\tilde{\alpha}^2_{ab,\sigma}}\right) \\
&\quad \times \Phi\left(\frac{\tilde{\beta}_{cd,\sigma}\tilde{\alpha}^2_{ab,\sigma} - \tilde{\beta}_{ab,\sigma}(\boldsymbol{\alpha}_{cd} C_\sigma \boldsymbol{\alpha}_{ab})}{\tilde{\alpha}_{ab,\sigma}\sqrt{\tilde{\alpha}^2_{cd,\sigma}\tilde{\alpha}^2_{ab,\sigma} - (\boldsymbol{\alpha}_{cd} C_\sigma \boldsymbol{\alpha}_{ab})^2}}\right) \\
&\quad + \frac{(C_\sigma \boldsymbol{\alpha}_{cd})_n}{\sqrt{(2\pi)}\tilde{\alpha}_{cd,\sigma}} \exp\left(-\frac{1}{2}\frac{\tilde{\beta}^2_{cd,\sigma}}{\tilde{\alpha}^2_{cd,\sigma}}\right) \\
&\quad \times \Phi\left(\frac{\tilde{\beta}_{ab,\sigma}\tilde{\alpha}^2_{cd,\sigma} - \tilde{\beta}_{cd,\sigma}(\boldsymbol{\alpha}_{ab} C_\sigma \boldsymbol{\alpha}_{cd})}{\tilde{\alpha}_{cd,\sigma}\sqrt{\tilde{\alpha}^2_{cd,\sigma}\tilde{\alpha}^2_{ab,\sigma} - (\boldsymbol{\alpha}_{cd} C_\sigma \boldsymbol{\alpha}_{ab})^2}}\right) \\
&\quad + (\boldsymbol{\mu}_\sigma)_n \langle \Theta_{ab} \Theta_{cd} \rangle_\sigma . \quad \text{(B.14)}
\end{aligned}$$

References

[1] Bibliography on the Self Organising Map (SOM) and Learning Vector Quantization (LVQ), Neural Networks Research Centre, Helsinki University of Technology, 2002.

[2] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.

[3] M. Biehl, A. Freking, A. Ghosh and G. Reents, A theoretical framework for analysing the dynamics of LVQ, Technical Report, Technical Report 2004-09-02, Mathematics and Computing Science, University Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands, December 2004, available from *www.cs.rug.nl/∼biehl*.

[4] M. Biehl, A. Freking and G. Reents, Dynamics of On-line Competitive Learning Europhysics Letters, 38: 73-78, 1996.

[5] M. Biehl, A. Ghosh and B. Hammer, Learning Vector Quantization: The Dynamics of Winner-Takes-All Algorithms. Neurocomputing, 69: 660-670, 2006.

[6] M. Biehl, A. Ghosh, and B. Hammer, Dynamics and Generalization Ability of LVQ Algorithms. Journal of Machine Learning Research, (8): 323-360 (2007)

[7] J. Buhmann, Stochastic Algorithms for Exploratory Data Analysis: Data Clustering and Data Visualization, in: Learning in Graphical Models, M. Jordan (ed.), Kluwer, 1997.

[8] A. Engel and C. van Broeck. The Statistical Mechanics of Learning, Cambridge University Press, 2001

[9] A. Gersho and R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Press, 1992.

[10] A. Ghosh, M. Biehl and B. Hammer, Performance Analysis of LVQ Algorithms: A Statistical Physics Approach, Neural Networks, special issue on Advances in Self-Organizing Maps. Vol. 19:817-829, 2006.

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

[12] T. Kohonen. Self Organising Maps, Springer, 3rd ed., 2001

[13] T. Martinetz, S. Berkovich, K. Schulten, 'Neural Gas' network for vector quantization and its application to time series prediction, IEEE TNN, 4(4):558-569, 1993.

[14] M.N. Murty, A.K. Jain, P.J. Flynn, Data clustering: a review, ACM Comput. Surveys 31 (1999) 264-323.

[15] G. Reents and R. Urbanczik, Self Averaging and On-line Learning, Phys. Rev. Letter, 80:5445-5448, 1998.

[16] T.L.H. Watkin, A. Rau, M. Biehl, The statistical mechanics of learning a rule, Rev. Mod. Phys. 65 (1993) 499-556.

[17] A. Witoelar, M. Biehl, A. Ghosh and B. Hammer, On the Dynamics of Vector Quantization and Neural Gas. In M. Verleysen, editor, European Symposium on Artificial Neural Networks, ESANN '07, d-side, Evere, Belgium, 127-132, 2007.

Aree Witoelar is currently a Ph.D. candidate in the Intelligent Systems Group of the Institute of Mathematics and Computing Science in University of Groningen, the Netherlands. He received his B.S. Degree in Engineering Physics from Bandung Institute of Technology, Indonesia, in 2002 and his M.Sc. Degree in Physics from University of Groningen, the Netherlands in 2005. His research interest is in the theory and application of machine learning, pattern recognition, clustering and self-organising maps.

Michael Biehl received a Ph.D. in Theoretical Physics from the University of Giessen, Germany, in 1992 and the venia legendi in Theoretical Physics from the University of Wuerzburg, Germany, in 1996. In 2003 he was appointed Assistant Professor in Computing Science at the University of Groningen, The Netherlands. His main research interest is in the theory, modelling and application of Machine Learning techniques. He is furthermore active in the modelling and simulation of complex physical systems. He has co-authored more than 70 papers in international journals and conferences; preprint versions and further information can be obtained from http://www.cs.rug.nl/ biehl/

Anarta Ghosh received his B.Sc (1995) with honours is Physics (first class) from University of Calcutta, India. He obtained the Master of Engineering (2000) degree (first class) in Electrical Engineering from Indian Institute of Science, Bangalore. After a brief stint as a digital signal processing engineer at Lucent Technologies (Agere Systems) he joined Institute of Mathematics and Computing Science, University of Groningen, The Netherlands as a Ph.D student. After getting the Ph.D degree in the field of computer vision and machine learning in January 2007, presently he is working as a senior fellow at University of Washington, Seattle, USA. His research interests are in the areas of machine learning, computer vision, computational/mathematical modeling, pattern recognition and signal/image processing.

Barbara Hammer received her Ph.D. in Computer Science in 1995 and her venia legendi in Computer Science in 2003, both from the University of Osnabrueck, Germany. From 2000-2004, she was leader of the junior research group 'Learning with Neural Methods on Structured Data' at University of Osnabrueck before accepting an offer as professor for Theoretical Computer Science at Clausthal University of Technology, Germany, in 2004. Several research stays have taken her to Italy, U.K., India, France, and the U.S.A. Her areas of expertise include machine learning, hybrid systems, self-organizing maps, clustering, recurrent networks, bioinformatics, and cognitive science.