# Vector-Quantization using Information Theoretic Concepts

Tue Lehn-Schiøler (tls@imm.dtu.dk)
*Intelligent Signal Processing, Informatics and Mathematical Modelling, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark*

Anant Hegde (ahegde@cnel.ufl.edu), Deniz Erdogmus (deniz@cnel.ufl.edu) and Jose C. Principe (principe@cnel.ufl.edu)
*Computational NeuroEngineering Laboratory, Electrical & Computer Engineering Department, University of Florida, Gainesville, FL 32611, USA*

**Abstract.** The process of representing a large data set with a smaller number of vectors in the best possible way, also known as vector quantization, has been intensively studied in the recent years. Very efficient algorithms like the Kohonen Self Organizing Map (SOM) and the Linde Buzo Gray (LBG) algorithm have been devised. In this paper a physical approach to the problem is taken, and it is shown that by considering the processing elements as points moving in a potential field an algorithm equally efficient as the before mentioned can be derived. Unlike SOM and LBG this algorithm has a clear physical interpretation and relies on minimization of a well defined cost-function. It is also shown how the potential field approach can be linked to information theory by use of the Parzen density estimator. In the light of information theory it becomes clear that minimizing the free energy of the system is in fact equivalent to minimizing a divergence measure between the distribution of the data and the distribution of the processing element, hence, the algorithm can be seen as a density matching method.

**Keywords:** Information particles, Information theoretic learning, Parzen density estimate, Self organizing map, Vector-Quantization

**Abbreviations:** SOM – Self-organized map; PE – Processing element; C-S – Cauchy-Schwartz; K-L – Kullback-Leibler; VQIT – Vector-Quantization using Information Theoretic Concepts; QE – Quantization error LBG – Linde Buzo Gray

## 1. Introduction

The idea of representing a large data set with a smaller set of processing elements (PE's) has been approached in a number of ways and for various reasons. Reducing the number of data points is vital for computation when working with a large amount of data whether the goal is to compress data for transmission or storage purposes, or to apply a computationally intensive algorithm.

In vector quantization, a set of data vectors is represented by a smaller set of code vectors, thus requiring only the code vector to be stored or transmitted. Data points are associated with the nearest code vector generating a lossy compression of the data set. The challenge is

to find the set of code vectors (the code book) that describes data in the most efficient way. A wide range of vector quantization algorithms exist, the most extensively used are K-means (MacQueen, 1967) and LBG (Linde et al., 1980).

For other applications like visualization, a good code book is not enough. The 'code vectors', or processing elements (PE's), as they are often denoted in the self-organizing literature, must preserve some predefined relationship with their neighbors. This is achieved by incorporating competition and cooperation (soft-competition) between the PE's. Algorithms with this property create what is known as Topology Preserving Maps. The Self-Organized Map (SOM) (Kohonen, 1982) is the most famous of these. It updates not only the processing element closest to a particular data point, but also its neighbors in the topology. By doing this it aligns the PE's to the data and at the same time draws neighboring PE's together. The algorithm has the ability to 'unfold' a topology while approximating the density of the data.

It has been shown (Erwin et al., 1992) that when the SOM has converged, it is at the minimum of a cost function. This cost-function is highly discontinuous and drastically changes if any sample changes its best matching PE. As a result it is not possible to use the conventional methods to optimize and analyze it. Further more, the cost function is not defined for a continuous distribution of input points since boundaries exist where a sample could equally be assigned to two different PE's (Erwin et al., 1992). Attempts has been made to find a cost function that, when minimized, gives results similar to the original update rule (Heskes and Kappen, 1993).

Efforts have also been made to use information theoretic learning to find good vector quantifiers and algorithms for Topology Preserving Maps. Heskes (1999) introduces a cost function as a free energy functional consisting of two parts, the quantization error and the entropy of the distribution of the PE's. He also explored the links between SOM, vector quantization, Elastic nets (Durbin and Willshaw, 1987) and Mixture modeling, concluding that the methods are closely linked via the free energy. Van Hulle (2002) uses an information theoretic approach to achieve self-organization. The learning rule adapts the mean and variance of Gaussian kernels to maximize differential entropy. This approach, however, leads to a trivial solution where PE's eventually coincide. To circumvent this, Van Hulle proposes to maximize the differential entropy and at the same time minimize the mutual information by introducing competition between the kernels. The competition is not based on information theory but rather implements an activity-based, winner-takes all heuristic. Bishop et al. (1996) proposes an algorithm

(the Generative Topographic Map) in which a mapping between a lattice of PE's and data space is trained using the EM algorithm.

Ideas on interactions between energy particles have been explored previously by Scofield (1988). However, in this paper, we approach the same problem with an information-theory perspective and explicitly use the probability distributions of the particles to minimize the free energy between them.

In this paper, an algorithm for vector quantization using information theoretic learning (VQIT) is introduced. Unlike the methods described above, this algorithm is designed to take the distribution of the data explicitly into account. This is done by matching the distribution of the PE's with the distribution of the data. This approach leads to the minimization of a well defined cost function. The central idea is to minimize the free energy of an information potential function. It is shown that minimizing free energy is equivalent to minimizing the divergence between a Parzen estimator of the PE's density distributions and a Parzen estimator of the data distribution. In section 2, an energy interpretation of the problem is presented and it is shown how this has close links to information theory. In section 3, the learning algorithm is derived using the Cauchy-Schwartz inequality. Numerical results are presented in section 4, where performance is evaluated on an artificial data set. In section 5 limitations and possible extensions to the algorithm are discussed and it is compared to already existing algorithms. Finally, concluding remarks are given in section 6.

## 2. Energy interpretation

The task is to choose locations for the PE's, so that they represent a larger set of data points as efficiently as possible. Consider two kind of particles; each kind has a potential field associated with it, but the polarity of the potentials are opposite. One set of particles (the data points) occupies fixed locations in space while the other set (the PE's) are free to move. The PE's will move according to the force exerted on them by data points and other PE's, eventually minimizing the free energy. The attracting force from data will ensure that the PE's are located near the data-points and repulsion between PE's will ensure diversity.

The potential field created by a single particle can be described by a kernel of the form $K(\cdot)$. Placing a kernel on each particle, the potential

energy at a point in space $\mathbf{x}$ is given by

$$p(\mathbf{x}) = \frac{\mathbf{1}}{\mathbf{N}} \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{K}(\mathbf{x} - \mathbf{x_i}) \tag{1}$$

where the index $i$ runs over the positions of all particles ($\mathbf{x_i}$) of a particular charge. If the kernel decays with distance ($K(x) \propto \frac{1}{(\mathbf{x}-\mathbf{x_i})}$) the potential is equivalent to physical potentials like gravitation and electric fields. However, in the information theoretic approach, any symmetric kernel with maximum at the center can be chosen. For the sake of simplicity, Gaussian kernels are used herein.
Due to the two different particle types, the energy of the system has contributions from three terms:

1. Interactions between the data points; since the data points are fixed, these interactions will not influence minimization of the energy.

2. Interactions between the data and the processing elements; due to the opposite signs of the potentials, these particles will attract each other and hence maximize correlation between the distribution of data and the distribution of PE's.

3. Interactions between PE's; the same sign of all the PE's potentials causes them to repel each other.

In the information theoretic literature equation (1) is also considered a density estimator. In fact it is exactly the well known Parzen density estimator (Parzen, 1962). In order to match the PE's with the data, we can use equation (1) to estimate their densities and then minimize the divergence between the densities. The distribution of the data points ($x_i$) can be written as $f(x) = \sum_i G(x - x_i, \sigma_f)$ and the distribution over PE's ($w_i$) as $g(x) = \sum_i G(x - w_i, \sigma_g)$.

Numerous divergence measures exist, of which the Kullback-Leibler (K-L) divergence is the most commonly used (Kullback and Leibler, 1951). The Integrated square error and the Cauchy-Schwartz (C-S) inequality, are both linear approximations to the K-L divergence. If C-S is used, the link between divergence and energy interpretation becomes evident. The Cauchy-Schwartz inequality,

$$|ab| \leq ||a||||b|| \tag{2}$$

is an equality only when vectors $a$ and $b$ are collinear. Hence, maximizing $\frac{|ab|}{||a||||b||}$ is equivalent to minimizing the divergence between $a$ and $b$. To remove the division, the logarithm can be maximized instead. This

is valid since the logarithm is a monotonically increasing function. In order to minimize the divergence between the distributions $f(x)$ and $g(x)$ the following expression is minimized:

$$D_{c-s}(f(x), g(x)) = -\log \frac{(\int (f(x)g(x))\mathrm{d}x)^2}{\int f^2(x)\mathrm{d}x \int g^2(x)\mathrm{d}x} \qquad (3)$$
$$= \log \int f^2(x)\mathrm{d}x - 2\log \int f(x)g(x)\mathrm{d}x + \log \int g^2(x)\mathrm{d}x$$

Following Principe et al. (2000) $V = \int g^2(x)\mathrm{d}x$ is denoted as the information potential of the PE's and $C = \int f(x)g(x)\mathrm{d}x$ the cross information potential between the distributions of data and the PE's. Note that

$$H(x) = -\log \int g^2(x)\mathrm{d}x = -logV \qquad (4)$$

is exactly the Renyi quadratic entropy (Rényi, 1970) of the PE's. As a result, minimizing the divergence between $f$ and $g$ is equal to maximizing the sum of the entropy of the PE's and the cross information potential between the densities of the PE's and the data. The link between equation (3) and the energy formulation can be seen by comparing the terms with the items in the list above.

## 3. The algorithm

As described in the previous section, finding the minimum energy location of the PE's in the potential field is equivalent to minimizing the divergence between the Parzen estimate of the distribution of data points $f(x)$ and the estimator of the distribution of the PE's $g(x)$. The Parzen estimate for the data has a total of $N$ kernels, where $N$ is the number of data points, and the Parzen estimator for the PE's uses $M$ kernels, $M$ being the number of processing elements (typically $M << N$).

Any divergence measure can be chosen, but in the following the derivation will be carried out for the Cauchy-Schwartz divergence,

$$J(w) = \log \int f^2(x)\mathrm{d}x - 2\log \int f(x)g(x)\mathrm{d}x + \log \int g^2(x)\mathrm{d}x \qquad (5)$$

The cost function $J(w)$ is minimized with respect to the location of the PE's $(w)$.

When the PE's are located such that the potential field is at a local minima, no effective force acts on them. Moving the PE's in the opposite direction of the gradient will bring them to such a potential

minimum; this is also known as the gradient descent method. The derivative of equation (5) with respect to the location of the PE's must be calculated; but, since the data points are stationary only the last two terms of equation (5) (the cross information potential and the entropy of the PE's) have non-zero derivatives.

For simplicity, the derivation of the learning rule has been split in two parts; calculation of the contribution from cross information potential and calculation of the contribution from entropy. In the derivation Gaussian kernels are assumed, although, any symmetric kernel that obeys Mercer's condition (Mercer, 1909) can be used.
Consider the cross information potential term ($\log \int f(x)g(x)\mathrm{d}x$); the Parzen estimator for $f(x)$ and $g(x)$ puts Gaussian kernels on each data point $x_j$ and each PE $w_i$ respectively, where the variances of the kernels are $\sigma_f^2$ and $\sigma_g^2$. Initially the location of the PE's are chosen randomly.

$$
\begin{aligned}
C &= \int \hat{f}(x)\hat{g}(x)\mathrm{d}x && \text{(6a)}\\
&= \frac{1}{MN}\int \sum_i^M G(x-w_i,\sigma_g^2)\sum_j^N G_f(x-x_j,\sigma_f^2)\mathrm{d}x && \text{(6b)}\\
&= \frac{1}{MN}\sum_i^M\sum_j^N \int G(x-w_i,\sigma_g^2)G(x-x_j,\sigma_f^2)\mathrm{d}x && \text{(6c)}\\
&= \frac{1}{MN}\sum_i^M\sum_j^N G(w_i-x_j,\sigma_a^2) && \text{(6d)}
\end{aligned}
$$

where the covariance of the Gaussian after integration is $\sigma_a^2 = \sigma_f^2 + \sigma_g^2$. The gradient update for PE $w_k$ from the cross information potential term then becomes:

$$\frac{\mathrm{d}}{\mathrm{d}w_k} 2\log C = -2\frac{\Delta C}{C} \tag{7}$$

Where $\Delta C$ denotes the derivative of $C$ with respect to $w_k$.

$$\Delta C = -\frac{1}{MN}\sum_j^N G_a(w_k-x_j,\sigma_a)\sigma_a^{-1}(w_k-x_j) \tag{8}$$

Similarly for the entropy term($-\log \int g^2(x)\mathrm{d}x$)

$$
\begin{aligned}
V = \int \hat{g}^2(x)\mathrm{d}x &= \frac{1}{M^2}\sum_i^M\sum_j^M G(w_i-w_j,\sqrt{2}\sigma_g) && \text{(9a)}\\
\frac{\mathrm{d}}{\mathrm{d}w_k}\log V &= \frac{\Delta V}{V} && \text{(9b)}
\end{aligned}
$$

With

$$\Delta V = -\frac{1}{M^2}\sum_{i}^{M} G(w_k - w_i, \sqrt{2}\sigma_g)\sigma_g^{-1}(w_k - w_i) \tag{10}$$

The update for point $k$ consist of two terms; cross information potential and entropy of the PE's:

$$w_k(n+1) = w_k(n) - \eta\left(\frac{\Delta V}{V} - 2\frac{\Delta C}{C}\right) \tag{11}$$

where $\eta$ is the step size. The final algorithm for vector-quantization using information theoretic concepts (VQIT), consist of a loop over all $w_k$. Note that $\Delta C$ and $\Delta V$ are directional vectors where as $C$ and $V$ are scalar normalizations.

As with all gradient based methods this algorithm has problems with local minima. One of the ways local minima can be avoided is by annealing the kernel size (Erdogmus and Principe, 2002). The potential created by the particles will depend on the width of the kernels and the distance between the particles. When the distance is large compared to the width, the potential will be very 'bumpy' and have many local minima, and when the particles are close compared to the width, the corresponding potential will be 'smooth'. If, in addition, the number of particles is large the potential will have the shape of a normal distribution. Starting with a large kernel size will therefore help to avoid local minima. As with the SOM, a good starting point is to choose the kernels such that all particles interact with each other.

The algorithm derived in this section uses the gradient descent method to minimize an energy function based on interactions between information particles. Each iteration of the algorithm requires $\mathcal{O}(M^2N)$ Gaussian evaluations due to the calculation of $C$ for each PE. The parameters for the algorithm are the variances of the density estimators $\sigma_f^2$ and $\sigma_g^2$ along with the step size $\eta$. The variances can be set equal and can be annealed from a size where all particles interact. The step size should be chosen small enough to ensure smooth convergence.

## 4. Simulations

In this section the ability of the VQIT algorithm to perform vector quantization is illustrated on a synthetic data set consisting of two half circles with unit radius which has been distorted with Gaussian noise with variance 0.1. One of the halves is displaced in horizontal direction (Figure 1).

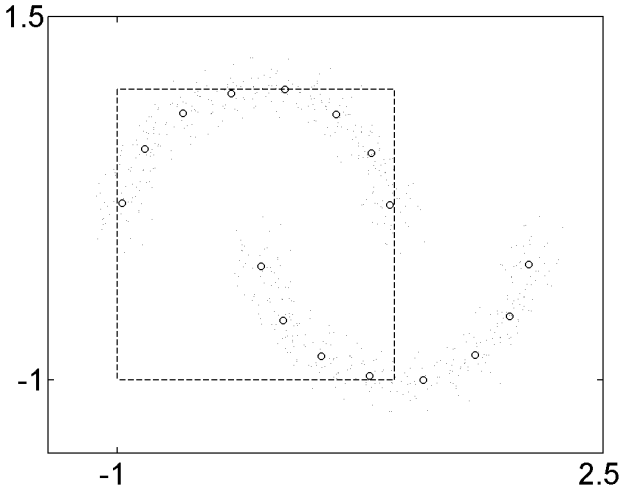
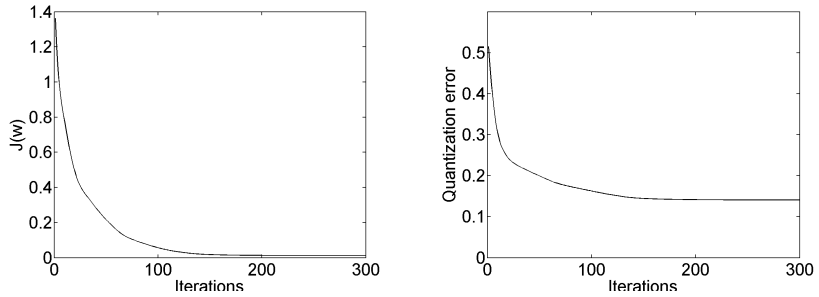



*Figure 1.* Artificial data used to evaluate performance, points are chosen from two half circles distorted by Gaussian noise. Initially all processing elements (PE's) were chosen randomly from the unit square, in all simulations the algorithm converged to the same solution (indicated by circles).

The data essentially consist of two clusters, as shown in Figure 1. Initially, 16 PE's are placed at random locations. The objective is to have the 16 PE's efficiently capture the structural property of the data.




a. Development of the cost-function averaged over 50 trials. The cost-function is always non-negative and has its minimum at zero but it is not guaranteed that a cost of zero can be achieved.

b. The quantization error measure is included for comparison with other algorithms.

*Figure 2.* Convergence of the algorithm, cost-function and quantization error.

Table I. Mean square errors for the data set shown in figure 1, the results are the average of 50 trials with different initial conditions. The Som, LBG and the VQIT algorithm always converges to the same solution.

| | VQIT | SOM | LBG | K-means |
|---|---|---|---|---|
| QE | 0.1408 | 0.1419 | 0.1393 | 0.1668 |

Using the algorithm presented above, the final locations of the PE's are shown, all in proper alignment with the data (Figure 1).

To assess the convergence of the VQIT, 50 monte-carlo simulations were performed. Starting with different initial conditions chosen uniformly from the unit square, it was found that with the right choice of parameters the algorithm always converges to the same solution. During training mode, having an initial large kernel-size and progressively annealing it can avoid the local minima. In this simulation, the width of the kernels was adjusted to equal the data-variance on each of its individual projections. The initial kernel size for the PE's was set to be:

$$\sigma_g = \sigma_n \begin{bmatrix} 0.75 & 0 \\ 0 & 0.5 \end{bmatrix}$$

where $\sigma_n$ is the decaying variable. This is initially set to $\sigma_0 = 1$ and it decays after every iteration according to:

$$\sigma_n = \frac{\sigma_0}{1 + (0.05\sigma_0 n)}$$

The kernel size for the data ($\sigma_f$) was set equal to $\sigma_g$.

The evolution of the cost-function is shown in figure 2.a. Note that the cost-function is always positive and that the minimum value it can obtain is zero. The quantization error (QE) is also calculated by computing the average distance between the data points and their corresponding winner PE. The QE convergence curve is shown in figure 2.b. To compare with other algorithms, the quantization error is used as a figure of merit since it is a commonly used evaluation metric. Comparison is provided with three algorithms: SOM, LBG and K-means. K-means is the only algorithm of these that does not converge to the same solution regardless of initial conditions. The result of the comparison can be seen in Table I. The quantization error for the VQIT, SOM, and LBG centers around 0.14 while the K-means does not perform as well. It should be noted that none of the algorithms directly minimizes QE, however, LBG includes it in the iterations.

## 5. Discussion

In this section some of the critical issues regarding the algorithm are discussed. Emphasis is put on links to other algorithms and possible extensions.

The algorithm presented in this work is derived on the basis of the Cauchy-Schwartz inequality. This leads to a divergence measure based on the inner-product between two vectors in a Riemann space. As noted earlier this is not the only choice, and has infact only been used here because of its close links to entropy. Another choice for cost-function is the Integrated Square Error which uses the quadratic distance between the distributions instead of an inner product:

$$\int (f(x) - g(x))^2 \mathrm{d}x = \int f^2(x)\mathrm{d}x - 2\int f(x)g(x)\mathrm{d}x + \int g^2(x)\mathrm{d}x. \quad (12)$$

The terms correspond to the information potentials of the data and the PE's and the cross information potential between the two. Note that equation (12) is similar to equation (5) except for the logarithm. Derivations equivalent to those used for C-S yields the very simple update:

$$w_k = w_k + \eta\,(\Delta V - \Delta C) \quad (13)$$

which requires $\mathcal{O}(MN)$ calculations per iteration. Annealing can also be used and the performance is similar to the VQIT.

"Density estimation is an ill posed problem and requires large amount of data to solve well" (Vapnik, 1995). Therefore, Vapnik suggests that one should not try to estimate densities in order to solve simpler problems (like vector quantization).

Even though this approach uses Parzen density densimates in its formulation, the pdf is never estimated. Instead the integral can be computed exactly through the double sum and therefore the method does not violate Vapnik's recommendations.

In a physical system, all potentials have the same form and only the magnitude (charge) can change, i.e. the same kernel type must be used for all particles. Also, in the Parzen estimator the mixture is homoskedastic, i.e. all mixtures have the same variance. However, in many of the recent publications (Van Hulle, 2002, Yin and Allinson, 2001, Heskes, 1999), a heteroskedastic approach is followed allowing the variance and weighting of the mixture components to change. It is easy to extend the algorithm presented in this work to include heteroskedastic

mixtures. The cost-function can just as well be minimized with respect to both means, variances and mixture weights. One can then choose to use either gradient descent or the EM algorithm to find the minimum. However, introducing more free parameters also means estimating more parameters from the same data points and can therefore lead to over fitting and poor generalization performance.

Another important issue is topology preservation. This feature is important if the mapping is to be used for visualization. Unlike the SOM, the learning rule proposed in this work is not topology preserving; it does not define an ordering of the PE's. It is however important to notice that if an ordering exists, the algorithm will approximately keep this ordering during convergence. Two different alterations can ensure that neighbors in the topology are also neighbors in the mapping.
The first and simplest is to add a term to the cost function equation (5). The term should include attraction from PE's that are close on the grid, one possibility is:

$$\sum_{i \in \mathcal{N}} (x_j - x_i) \tag{14}$$

Where $\mathcal{N}$ is the set of neighbors defined by the topology. Since the cost-function is changed, we cannot expect the PE's to converge to the same positions. However, once the topology has unfolded, the weighting of the neighborhood term equation (14) can be reduced and a solution will be obtained with PE at the desired positions and this time with the desired topology.
Another option more along the lines of the SOM and other algorithms (Yin and Allinson, 2001, Van Hulle, 2002), is to change the update of the cross information potential term. If we chose a winner PE for every data point and then update only itself and its neighbors, PE's close in the topology will be drawn together. Unfortunately this is not straight forward to put into the information theoretic framework.

The VQIT algorithm is based on block-computation of the data. It is possible to develop an online sample-by-sample update, which may result in a significant reduction in computational complexity. One way this can be achieved is by eliminating the second summation in equation (6) and computing the Kernel for only the current sample. However, this idea is still being explored and efforts directed at finding its similarity with the Kohonen-SOM will be addressed in a future paper.

## 6. Conclusion

In this paper an algorithm for finding the optimal quantization of a data set is proposed. The algorithm is derived based on concepts from information theoretic learning and it is shown how information potential fields and Parzen estimators can be used to give a physical interpretation of vector quantization. Simulations show errors equivalent to those obtained by the SOM and the LBG algorithms. However, unlike SOM and LBG, the algorithm proposed here utilizes a cost-function and its derivative. The algorithm can easily be extended to form a topology preserving map.

Future efforts will be directed towards comparing numerical properties of the algorithm and to incorporate the suggested changes. Further more, it will be interesting to see how VQIT performs on real data.

The main contribution of this work is a novel approach to vector-quantization utilizing physical laws and introducing probability densities directly into the optimization.

Acknowledgement

This work is partially supported by NSF grant ECS- 0300340

## References

Bishop, C. M., M. Svensen, and C. K. I. Williams: 1996, 'GTM: a principled alternative to the self-organizing map'. *Artificial Neural Networks—ICANN 96. 1996 International Conference Proceedings* pp. 165–70.

Durbin, R. and D. Willshaw: 1987, 'An Analogue Approach of the Travelling Salesman Problem Using an Elastic Net Method'. *Nature,* **326**, 689–691.

Erdogmus, D. and J. C. Principe: 2002, 'Generalized Information Potential Criterion for Adaptive System Training'. *IEEE Transactions on Neural Networks* **13**(5).

Erdogmus, D., J. C. Principe, and K. Hild: 2002, 'Beyond second-order statistics for learning'. *Natural Computing* **1**(1), 85–108.

Erwin, E., K. Obermayer, and K. Schulten: 1992, 'Self-organizing maps: ordering, convergence properties and energy functions'. *Biological Cybernetics* **67:4755**.

Graepel, T., M. Burger, and K. Obermeyer: 1995, 'Phase Transitions in Stochastic Sef-Organizing Maps'. *Physical Review E* **56**(4), 3876–3890.

Heskes, T.: 1999, 'Energy functions for self-organizing maps'. In: S. E. Oja and Kaski (eds.): *Kohonen Maps.* Amsterdam: Elsevier, pp. 303–316.

Heskes, T. and B. Kappen: 1993, 'Error potentials for self-organization'. *Proceedings IJCNN93* **3**, 1219–1223.

Kohonen, T.: 1982, 'Self-organized formation of topologically correct feature maps'. *Biol. Cybern.* **43**, 59–69.

Kullback, S. and R. A. Leibler: 1951, 'On information and sufficiency'. *The Annals of Mathematical Statistics* **22**, 79–86.

Lampinen, J. and T. Kostiainen: 2002, 'Generative Probability Density Model in the SelfOrganizing Map'.

Linde, Y., A. Buzo, and R. M. Gray: 1980, 'An algorithm for vector quantizer design'. *IEEE Trans Commun COM* **28**, 84–95.

MacQueen, J.: 1967, 'Some methods for classification and analysis of multivariate observations'. *Proceedings of the Fifth Berkeley Symposium on Mathematical statistics and probability* **1**, 281–297.

Mercer, J.: 1909, 'Functions of positive and negative type and their connection with the theory of integral equations'. *Philosophical Transactions Royal Society London, A* **209**, 415–446.

Parzen, E.: 1962, 'On estimation of a probability density function and mode'. *Ann. Math. Stat* **27**, 1065–1076.

Principe, J. C., D. Xu, Q. Zhao, and J. Fisher: 2000, 'Learning from examples with information theoretic criteria'. *Journal of VLSI Signal Processing-Systems* **26**(1-2), 61–77.

Rényi, A.: 1970, *Probability Theory*. North-Holland Publishing Company, Amsterdam.

Scofield, C. L.: 1988, 'Unsupervised learning in the N-dimensional Coulomb network'. *Neural Networks* **1**(1), 129.

Sum, J., C.-S. Leung, L.-W. Chan, and L. Xu: 1997, 'Yet another algorithm which can generate topography map'. *Neural Networks, IEEE Transactions on* **8**(5), 1204–1207.

Van Hulle, M. M.: 2002, 'Kernel-based topographic map formation achieved with an information-theoretic approach'. *Neural Networks* **15**(8-9), 1029–1039.

Vapnik, V. N.: 1995, *The Nature of Statistical Learning Theory*. Springer-Verlag, New-York.

Yin, H. and N. Allinson: 2001, 'Self-organizing mixture networks for probability density estimation'. *Neural Networks, IEEE Transactions on,* **12**(2), 405–411.

*Address for Offprints:* Tue Lehn-Schiøler
Intelligent Signal Processing
Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads
DTU-Building 321
2800 Kgs. Lyngby
Denmark