Neural Computation no. 2465R

Joint entropy maximization in kernel-based topographic maps

Marc M. Van Hulle

K.U.Leuven, Laboratorium voor Neuro- en Psychofysiologie
Campus Gasthuisberg, Herestraat 49, B-3000 Leuven, BELGIUM
E-mail: marc@neuro.kuleuven.ac.be

March 8, 2002

Abstract

A new learning algorithm for kernel-based topographic map formation is introduced. The kernel parameters are adjusted individually so as to maximize the joint entropy of the kernel outputs. This is done by maximizing the differential entropies of the individual kernel outputs, given that also the map's output redundancy, due to the kernel overlap, needs to be minimized. The latter is achieved by minimizing the mutual information *between* the kernel outputs. As a kernel, the (radial) incomplete gamma distribution is taken since, for a Gaussian input density, the differential entropy of the kernel output will be maximal. Since the theoretically-optimal joint entropy performance can be derived for the case of non-overlapping Gaussian mixture densities, a new clustering algorithm is suggested that uses this optimum as its "null" distribution. Finally, it is shown that the learning algorithm is similar to one which performs stochastic gradient descent on the Kullback-Leibler divergence for a heteroscedastic Gaussian mixture density model.

1 Introduction

In an attempt to improve the density estimation properties, the noise tolerance, or even the biological relevance of the Self-Organizing Map (SOM) (Kohonen, 1982,1995), algorithms have been devised that can accommodate neurons with kernel-based activation functions, such as Gaussians, instead of Winner-Take-All (WTA) functions (Voronoi tessellation). An early example is the elastic net of Durbin and Willshaw (1987), which can be viewed as an equal-variance- or homoscedastic Gaussian mixture density model, fitted to the data points by a penalized maximum likelihood term. More recent examples of the density modeling approach are the algorithms introduced by Bishop and co-workers (Bishop *et al.*, 1998) (Generative Topographic Map, based on constrained, homoscedastic Gaussian mixture density modeling with equal mixings), Utsugi (1997) (also using equal mixings of homoscedastic Gaussians), and Van Hulle (1998) (equiprobabilistic maps using heteroscedastic Gaussian mixtures – thus, with differing variances). Furthermore, we should also mention the fuzzy membership in clusters approach of Graepel and co-workers (Graepel *et al.*, 1997), and the maximization of local correlations approach of Xu and co-workers (Sum *et al.*, 1997), both of which rely on homoscedastic Gaussians. Graepel and co-workers (Graepel *et al.*, 1998) also proposed a still different approach to kernel-based topographic map formation by introducing a non-linear transformation that maps the data points to a high-dimensional "feature" space, and that, in addition, admits a kernel function, such as a Gaussian with fixed variance, as in (kernel-based) Support Vector Machines (SVMs) (Vapnik, 1995). This idea was recently taken up again by András (2001), but with the purpose of optimizing the map's classification performance, by individually adjusting the kernel radii using a supervised learning algorithm. Finally, the original SOM algorithm itself has been regarded as an approximate way to perform homoscedastic Gaussian mixture density modeling by Utsugi (1997), Yin and

Allinson (2001), Kostiainen and Lampinen (2002), among others.

Linsker was among the first to develop topographic maps by optimizing an information-theoretic criterion (Linsker, 1989). He applied his principle of maximum information preservation (mutual information maximization *between input and output* – *infomax* for short), to a network of WTA neurons. The question is now: can this principle also be applied to kernel-based topographic maps? The obvious answer is to express the average mutual information integral in terms of the kernel output densities – or probabilities when they are discretized –, and adjust the kernel parameters individually so that the integral is maximized. However, such an approach rapidly becomes infeasible in practice: Linsker already needed to restrict himself to binary outputs in his WTA network, in order to facilitate the computation of the integral. A different information-theoretic approach is to minimize the Kullback-Leibler divergence (also called relative- or cross-entropy) between the true and the estimated input density, an idea that has been introduced for kernel-based topographic map formation by Benaim and Tomasini (1991), using homoscedastic Gaussians, and extended more recently by Yin and Allinson (2001) to heteroscedastic Gaussians.

We will introduce in this paper a new learning algorithm for kernel-based topographic map formation that adjusts the kernel parameters individually, and in such a manner that the joint entropy of the kernel outputs is maximized. We will derive our learning algorithm, and the optimization criterion behind it, in a bottom-up manner by starting with differential entropy maximization: when this is maximized for a given kernel, then the kernel's parameters will be optimally adapted in the sense that the mutual information between the kernel output and its input will maximal. The kernel output function and the learning rule for this case are derived in sections 2 and 3, respectively. As our kernel output function, we take the (radial) incomplete gamma distribution, since the

kernel's differential entropy is theoretically maximal when the input density is Gaussian. Section 4 starts with the observation that differential entropy maximization alone is not sufficient when there are multiple kernels in the map, since all kernels will eventually coincide. As a measure for kernel overlap, and thus for the map's output redundancy, the mutual information *between* the kernel outputs is taken. Maximizing joint entropy is then put forward as our optimization principle since it unifies both requirements: maximizing the differential entropies of the kernel outputs given that also the map's mutual information needs to be minimized. Section 5 then complements the learning rules with a neighborhood function so that topographic maps can be developed. Since we are able to derive the theoretically-optimal joint entropy performance for the case of an input distribution consisting of non-overlapping Gaussians, we will suggest a new clustering algorithm that uses this optimum as its "null" distribution in section 6. In section 7, we show that our learning algorithm is similar to one which performs stochastic gradient descent on the Kullback-Leibler divergence when heteroscedastic Gaussian mixtures are used. Finally, in section 8, we discuss the correspondence with other learning algorithms for kernel-based topographic map formation.

2 Kernel definition

Consider a formal neuron i, the output of which is, in response to an input $\mathbf{v} \in \Re^d$, $\mathbf{v} = [v_1, \ldots, v_d]$, described by a (localized) kernel K centered at $\mathbf{w}_i = [w_{i1}, ..., w_{id}]$. For simplicity's sake, we consider the kernel to be radially symmetrical around its center, $K(\mathbf{v}, \mathbf{w}_i, \sigma_i) \equiv K(\|\mathbf{v} - \mathbf{w}_i\|, \sigma_i)$, with σ_i the kernel radius. As motivated by Bell and Sejnowski (1995), the mutual information between the output $y_i \in \Re$ of neuron i and its input $\mathbf{v}$, $I(y_i, \mathbf{v})$, $\mathbf{v} \in \Re^d$, will be maximized when the differential entropy of its output, $H(y_i)$, is maximized. When assuming that the kernel output y_i has bounded sup-

port, $H(y_i)$ will be maximized when the output distribution is uniform. This is the case when the output distribution matches the cumulative distribution function (repartition function) of the input density. This will be our kernel definition, and its parameters will be adapted to the local input density with an incremental ("on-line") learning algorithm. We will first restrict ourselves to Gaussian input densities.

Assume a d-dimensional Gaussian with mean $[\mu_1, \ldots, \mu_d]$, and unit variance. The squared Euclidean distance to the center $x \stackrel{\Delta}{=} \sum_{j=1}^{d}(v_j - \mu_j)^2$, is known to obey the chi-squared distribution with $\theta = 2$ and $\alpha = \frac{d}{2}$ degrees of freedom (Weisstein, 1999):

$$p_{\chi^2}(x) = \frac{x^{\frac{d}{2}-1} \cdot \exp\left(-\frac{x}{2}\right)}{2^{\frac{d}{2}}\Gamma(\frac{d}{2})}, \tag{1}$$

for $0 \leq x < \infty$, and with $\Gamma(.)$ the gamma distribution. Hence, the distribution of the Euclidean distance to the center becomes: $p(r) = 2\,r\,p_{\chi^2}(r^2)$, with $r = \sqrt{x}$, following the fundamental law of probabilities. After some algebraic manipulations, and by generalizing to a Gaussian with standard deviation σ, we can write the distribution of the Euclidean distances as follows:

$$p(r) = \frac{2(\frac{r}{\sigma})^{d-1} \exp\left(-\frac{(\frac{r}{\sigma})^2}{2}\right)}{2^{\frac{d}{2}}\Gamma(\frac{d}{2})}. \tag{2}$$

The distribution is plotted in Fig. 1 (thick and thin continuous lines). The mean of r equals $\mu_r = \frac{\sqrt{2}\sigma\Gamma(\frac{d+1}{2})}{\Gamma(\frac{d}{2})}$, which can be approximated as $\sqrt{d}\sigma$, for d large, using the approximation of Graham and co-workers (1994); the second moment around zero equals $d\sigma^2$. The distribution $p(r)$ quickly approaches a Gaussian with mean μ_r and standard deviation $\frac{\sigma}{\sqrt{2}}$ when d increases: *e.g.*, the skewness and Fisher kurtosis are $8.07\ 10^{-2}$ and $8.71\ 10^{-3}$ for $d = 10$, respectively.

Finally, the kernel is defined in accordance with the cumulative distribution of

$p(r)$, which is the (complement of the) incomplete gamma distribution:

$$y_i = K(\mathbf{v}, \mathbf{w}_i, \sigma_i) = P(\frac{d}{2}, \frac{\|\mathbf{w}_i - \mathbf{v}\|^2}{2\sigma_i^2}) \equiv \frac{\Gamma(\frac{d}{2}, \frac{\|\mathbf{w}_i - \mathbf{v}\|^2}{2\sigma_i^2})}{\Gamma(\frac{d}{2})}, \tag{3}$$

which is plotted as a function of the Euclidean distance $r = \|\mathbf{w}_i - \mathbf{v}\|^2$, for $d = 1, \ldots, 10$ in Fig. 1 (thick and thin dashed lines). Note that since $K(.)$ only depends on the Euclidean distance, the obtained kernel in the input space is radially-symmetrical.

3 Kernel adaptation

We will now derive an "on-line" stochastic learning algorithm that adapts an individual kernel's parameters in such a way that the differential entropy of the kernel output is maximized. Consider first the adaptation of the kernel center. The entropy of the kernel output of neuron i can be written as:

$$H(y_i) = -\int_0^\infty p_{y_i}(x) \ln p_{y_i}(x)\ dx, \tag{4}$$

with $p_{y_i}(.)$ the kernel output density which, in turn, can be written as a function of the distribution of the Euclidean distance to the kernel center r:

$$p_{y_i}(y_i) = \frac{p_r(r)}{|\frac{\partial y_i}{\partial r}|}. \tag{5}$$

After substitution of the latter into eq. (4), we obtain:

$$H(y_i) = -\int_0^\infty p_r(r) \ln p_r(r)\ dr + \int_0^\infty p_r(r)\ \ln |\frac{\partial y_i(r)}{\partial r}|\ dr. \tag{6}$$

By performing gradient ascent on $H(.)$, we obtain the "on-line" learning rule for the kernel center:

$$\Delta \mathbf{w}_i = \eta_w \frac{\partial H}{\partial r} \frac{\partial r}{\partial \mathbf{w}_i} = \eta_w \frac{\mathbf{v} - \mathbf{w}_i}{\sigma_i^2}, \quad \forall i, \tag{7}$$

with η_w the learning rate, after some algebraic manipulations (see Appendix). In a similar manner, the learning rule for the kernel radius σ_i is obtained:

$$\Delta \sigma_i = \eta_\sigma \frac{\partial H}{\partial \sigma_i} = \eta_\sigma \frac{1}{\sigma_i} \left(\frac{\|\mathbf{v} - \mathbf{w}_i\|^2}{d\sigma_i^2} - 1 \right), \quad \forall i, \tag{8}$$

with η_σ the learning rate.

4 Joint entropy maximization

Maximizing differential entropy alone is not sufficient when there are multiple kernels in the map. This can be easily shown as follows. Consider a lattice of two neurons with kernel outputs y_1 and y_2. When eq. (7,8) would be used, *e.g.*, in the case of a Gaussian input density, then the two kernels will eventually coincide, the neural outputs will be maximally statistically dependent, and thus the map maximally redundant. We can formulate statistical dependency (or redundancy) in information-theoretic terms as the mutual information *between* the neuron outputs. Hence, we maximize the differential entropy given that we also need to minimize the mutual information, in order to cope with the kernel overlap. This dual goal is captured by maximizing the joint entropy of the neuron outputs:

$$H(y_1, y_2) = H(y_1) + H(y_2) - I(y_1, y_2), \tag{9}$$

with $H(y_1, y_2)$ the joint entropy, $H(y_1)$ and $H(y_2)$ the differential entropies, and $I(y_1, y_2)$ the mutual information. We will perform mutual information minimization heuristically by putting kernel adaptation in a competitive learning context: in this way, the winning neuron's kernel will decrease its range – in particular when it is strongly active – and, thus, decrease its overlap with the surrounding kernels. In addition, we will add a neighborhood function to the learning process, since we want to achieve topology-preserving lattices. The learning rules are derived in the next section.

5 Topographic map formation

Consider a lattice A of N neurons and corresponding incomplete gamma distribution kernels $K(\mathbf{v}, \mathbf{w}_i, \sigma_i), i = 1, \ldots, N$. We introduce an *activity-based* competition between the neurons, with the "winning" neuron defined as $i^* = \arg\max_{\forall i \in A} y_i$, rather than the more common (minimum Euclidean)

distance-based competition, $i^* = \arg\min_i \|\mathbf{w}_i - \mathbf{v}\|$, which is equivalent to our case only when all kernels have equal radii. We supply topological information by means of a neighborhood function Λ, for which we take a monotonously decreasing function of the lattice distance from the winner. We opt for a Gaussian neighborhood function:

$$\Lambda(i, i^*, \sigma_\Lambda) = \exp\left(-\frac{\|\mathbf{r}_i - \mathbf{r}_{i^*}\|^2}{2\sigma_\Lambda^2}\right), \tag{10}$$

with σ_Λ the neighborhood function range, and $\mathbf{r}_i$ neuron i's lattice coordinate. The complete set of learning rules then becomes:

$$\Delta\mathbf{w}_i = \eta_w\ \Lambda(i, i^*, \sigma_\Lambda)\ \frac{\mathbf{v} - \mathbf{w}_i}{\sigma_i^2}, \tag{11}$$

$$\Delta\sigma_i = \eta_\sigma\ \Lambda(i, i^*, \sigma_\Lambda)\ \frac{1}{\sigma_i}\left(\frac{\|\mathbf{v} - \mathbf{w}_i\|^2}{d\sigma_i^2} - 1\right), \quad \forall i. \tag{12}$$

In case the neighborhood function would be omitted in eqs. (11,12), joint entropy maximization will still be aimed for, but the kernels will not become topographically organized, in particular when starting from a random initialization. Hence, one could envisage the purpose of the neighborhood function as a way to constrain the learning process so that it favors the development of topographic maps.

Lattice-disentangling dynamics

The disentangling dynamics is exemplified in Fig. 2 for the standard case of a square lattice and a uniform square input density. The weights were initialized by sampling from the same input density, the radii by sampling the uniform distribution $[0, 0.1]$. The neighborhood range was decreased as follows:

$$\sigma_\Lambda(t) = \sigma_{\Lambda 0}\ \exp\left(-2\sigma_{\Lambda 0}\frac{t}{t_{max}}\right), \tag{13}$$

with t the present time step, t_{max} the maximum number of time steps, and $\sigma_{\Lambda 0}$ the range spanned by the neighborhood function at $t = 0$. We took

$t_{max} = 2,000,000$ and $\sigma_{\Lambda 0} = 12$, so that the neighborhood function vanishes at the end of the learning process ($\sigma_\Lambda(t_{max} \approx 4.5 \times 10^{-10})$. We further took $\eta_w = 0.01$ and $\eta_\sigma = 10^{-4}\,\eta_w$.

6 Theoretically optimal and achieved performance

The theoretically optimal performance of the kernel-based map can be derived for two limiting cases. In the first case, we assume that the input density consists of N Gaussians that are spaced infinitely far apart so that their overlap is infinitesimally small ("N-Gaussians" case). We quantize the kernel outputs y_i, $\forall i \in A$, uniformly into k equally-sized and non-overlapping quantization intervals.[1] The optimal solution is reached when each Gaussian is modeled by a different kernel. The expressions for the joint entropy JE and mutual information MI can be derived analytically:

$$\text{JE} = \frac{k-1}{k}\log_2 kN + \frac{1}{k}\log_2 k, \tag{14}$$

$$\text{MI} = \frac{k-1}{k}\log_2\left(\frac{kN}{(N-1)k+1}\right)^{N-1} + \frac{1}{k}\log_2 \frac{\frac{1}{k}}{\left(\frac{(N-1)k+1}{kN}\right)^N}. \tag{15}$$

In Fig. 3A, JE and MI are plotted as a function of N and parameterized with respect to k. We can easily determine the asymptotes for MI when $k \to \infty$ (thin dashed line in Fig. 3A) and $k, N \to \infty$ (dot-dashed line):

$$\text{MI} = \log_2\left(\frac{N}{N-1}\right)^{N-1}, \quad \text{for } k \to \infty, \tag{16}$$

$$\text{MI} = \frac{1}{\log_e 2} \approx 1.4427, \quad \text{for } k, N \to \infty. \tag{17}$$

[1] Note that, as a result of this quantization, the aforementioned assumption is achieved as soon as, for each neuron, the tails of the $N-1$ other Gaussians activate only the neuron's lowest quantization interval, namely, the one that codes for the lowest kernel output values.

This means that, for a given number of neurons N, the mutual information will always be finite.[2] Furthermore, we also see that for any number of neurons and quantization levels, the mutual information will be bounded by the second asymptote. Finally, note that the optimal JE and MI do not depend on the input space dimensionality d.

The performance of our learning algorithm can be easily measured against these theoretical results. Consider a 1-dimensional lattice (chain) of N neurons, with $N = 1, \ldots, 10$, developed in the 1-dimensional input space. We take N unit-variance Gaussians, spaced 15 units apart along the real line. We initialize the weights and radii by sampling the $[0, 1]$ interval uniformly, and run our learning algorithm 20 times for several (N, k)-combinations. We take $t_{max} = 2,000,000$, $\sigma_{\Lambda 0} = 12$, $\eta_w = 0.01$ and $\eta_\sigma = 10^{-4}\,\eta_w$. The JE and MI plots obtained practically coincide with the optimal ones and are not shown in Fig. 3A: the differences between the theoretically optimal JE and MI values and the obtained averages are smaller than, respectively, 0.005 and 0.02 everywhere. This performance was maintained for input space dimensionalities $d = 1, \ldots, 10$ (Fig. 3C).

In the second case, we assume that the input density consists of one Gaussian only ("1-Gaussian" case). The expressions for the joint entropy and mutual information can be derived analytically, for the case of $k = 2$ quantization

[2] This can be intuitively understood by observing that only the lowest quantization interval accounts for all the overlap with the other kernels (see footnote 1) and that, as k increases, the activation probability of this interval becomes more and more "unique" with respect to the activation probabilities of the neuron's other quantization intervals (which form an equitable distribution). More specifically, the probability that the lowest quantization interval is active equals $\frac{(N-1)k+1}{kN}$, whereas that of the other intervals equals $\frac{1}{kN}$, from which follows that the mutual information will be bounded when $k \to \infty$.

levels and a 1-dimensional input space:

$$\begin{aligned} \text{JE} &= \log_2 2N, \\ \text{MI} &= \log_2 2N + 2\left(\frac{N-1}{N}\log_2\frac{N}{N-1} + \frac{1}{N}\right) + \\ &\quad (N-2)\left(\frac{3}{2N}\log_2\frac{2N}{3} + \frac{2N-3}{2N}\log_2\frac{2N}{2N-3}\right) + \log_2\frac{1}{2N}, \end{aligned} \quad (18)$$

with the second term in MI being present for $N > 2$ only. Figure 3B shows the JE and MI plots for different values of k; the plots for the $k > 2$ cases were determined numerically. Contrary to the N-Gaussians case, MI continues to increase when N increases. Furthermore, the optimal JE and MI plots depend on the input dimensionality d and are also determined numerically.

We also determine the performance of our learning algorithm for the 1-Gaussian case. The results are summarized in Fig. 3B. The average JE results (also for 20 runs) practically coincide with the theoretically-optimal ones (difference with optimal result < 0.02; standard deviation < 0.03). The results as a function of the input dimensionality d are shown in Fig. 3D for $k = 2$. The average JE plots again practically coincide with the numerically determined ones (difference with optimal result < 0.02; standard deviation < 0.025). The standard deviations on the MI results are similar to those shown in Fig. 3B, and are omitted for clarity's sake.

Finally, since we are using a heuristic for mutual information minimization (*i.e.*, competitive learning), we can only empirically verify the performance of the learning algorithm. We cannot formally prove that, for a given input distribution, the algorithm is guaranteed to converge towards a solution that will maximize the joint entropy. However, at least for the test cases considered here, we could verify that the achieved performance was satisfactory, since the theoretical results were available.

6.1 Clustering

The theoretical results suggest a new type of metric for determining the number of clusters in a data set, given that they can be appropriately modeled by Gaussians.[3] If the data set can be modeled by N Gaussians, that are spaced sufficiently far apart, then we know the theoretically optimal joint entropy (*cf.*, the N-Gaussians case). We also know that a mismatch in the number of neurons leads to a higher joint entropy: the JE values of the N-Gaussians case are always smaller than the corresponding values for the 1-Gaussian case (*e.g.*, $\frac{1}{2}\log_2 2N + \frac{1}{2} < \log_2 2N$, for $N > 1$, $k = 2$). Hence, the N-Gaussians case will be our "null" distribution, and the corresponding optimal joint entropy our reference value. We suggest the following clustering procedure:

for ($N \Leftarrow 1; N \leq$ `max_number_of_clusters`$; N + 1$)

develop a topographic map with N kernels and determine JE

determine the difference with JE eq. (14), assuming N Gaussians, one for each kernel

store the JE difference

select N with the smallest JE difference

The advantage of this procedure is twofold. First, we can still consider the one-cluster case, which is lacking in most clustering methods (Gordon, 1999). Second, our reference value is a theoretical result, instead of an estimate, as, *e.g.*, in the case of the Gap statistic, which looks at the point where the difference between two values of an intracluster distance metric becomes maximal, one for the given sample set and another for a reference set (Tibshirani *et al.*, 2001).

In order to test our procedure, we re-apply the benchmark Tibshirani and coworkers used for the Gap statistic. We consider 2 one-dimensional Gaussians

[3] Note that, in practice, single clusters can be appropriately modeled by log-concave functions, such as Gaussians.

with equal standard deviations, and vary the (Bayesian) overlap rate. For each overlap rate, we train our one-dimensional lattices in batch mode on 100 samples – 50 from each Gaussian –, and determine the configuration N for which JE is minimal, as explained above. We repeat this experiment 20 times, for different 100 sample sets, determine the probability that the correct number of clusters is found, and plot this probability as a function of the overlap rate. The result is shown in Fig. 4 (thick continuous line). We observe a marked drop in performance at about 18 % overlap rate, which is close to the point where the sum of two Gaussians becomes unimodal. The Gap statistic, using the Euclidean distance metric, is applied to the kernel centers, and also to the case where the weights are determined with the k-means clustering algorithm. The results are shown in Fig. 4 (thin continuous and dashed line, respectively). We now observe a gradual decrease in performance for an increasing overlap rate, until it drops below "chance" level (dot-dashed line), since the choice is in practice between one or two clusters.

7 Connection with Kullback-Leibler divergence

There is an interesting connection between joint entropy maximization, given our incomplete gamma distribution kernels, and the Kullback-Leibler divergence minimization, given heteroscedastic Gaussian kernels. The Kullback-Leibler divergence KL (also called relative- or cross-entropy) is a frequently used metric for assessing the quality of a density estimate. It is defined as follows:

$$\mathrm{KL} = -\int \log \frac{\widehat{p}(\mathbf{v})}{p(\mathbf{v})}\, p(\mathbf{v})\; d\mathbf{v}, \tag{19}$$

with $p(\mathbf{v})$ the true input density and $\widehat{p}(\mathbf{v})$ the estimated density. It is always a nonnegative number, and it will equal zero if and only if the density estimate is identical to the true density.

Assume that we perform a Gaussian mixture density modeling with equal mixings:

$$\widehat{p}(\mathbf{v}) = \frac{1}{N} \sum_{i=1}^{N} \frac{\exp\left(-\frac{\|\mathbf{v}-\mathbf{w}_i\|^2}{2\sigma_i{}^2}\right)}{(2\pi)^{\frac{d}{2}} \sigma_i^d}, \tag{20}$$

with $\mathbf{w}_i$ and σ_i the ith Gaussian's center and radius, respectively. The "optimal" density estimate is determined by minimizing KL. We need the partial derivatives with respect to the centers and radii:

$$\frac{\partial \mathrm{KL}}{\partial \mathbf{w}_i} = -\int \left(\frac{1}{\widehat{p}(\mathbf{v}|\Theta)} \frac{\partial \widehat{p}(\mathbf{v}|\Theta)}{\partial \mathbf{w}_i} \right) p(\mathbf{v})\, d\mathbf{v} = 0, \tag{21}$$

$$\frac{\partial \mathrm{KL}}{\partial \sigma_i} = -\int \left(\frac{1}{\widehat{p}(\mathbf{v}|\Theta)} \frac{\partial \widehat{p}(\mathbf{v}|\Theta)}{\partial \sigma_i} \right) p(\mathbf{v})\, d\mathbf{v} = 0, \quad \forall i, \tag{22}$$

with $\Theta = \{[\mathbf{w}_i], [\sigma_i]\}$, the parameter vector of the density model. Stochastic approximation (Robbins and Munro, 1951) can be invoked to solve these equations, which leads to the following learning rules:

$$\Delta \mathbf{w}_i = \eta_w\, \widehat{P}(\mathbf{v}|i)\, \frac{(\mathbf{v}-\mathbf{w}_i)}{\sigma_i^2}, \tag{23}$$

$$\Delta \sigma_i = \eta_\sigma\, \widehat{P}(\mathbf{v}|i)\, \frac{d}{\sigma_i} \left(\frac{\|\mathbf{v}-\mathbf{w}_i\|^2}{d\sigma_i^2} - 1 \right), \quad \forall i, \tag{24}$$

after some algebraic manipulations (see Appendix). The parameter $\widehat{P}(\mathbf{v}|i)$ represents the ith neuron's posterior probability. When we take for $\widehat{P}(\mathbf{v}|i) = \delta_{i\,i^*}$, with i^* the neuron that wins the competition, and introduce the conventional neighborhood function, since we wish to achieve a topology-preserving mapping, we obtain the following learning rules:

$$\Delta \mathbf{w}_i = \eta_w\, \Lambda(i, i^*, \sigma_\Lambda)\, \frac{(\mathbf{v}-\mathbf{w}_i)}{\sigma_i^2}, \tag{25}$$

$$\Delta \sigma_i = \eta_\sigma\, \Lambda(i, i^*, \sigma_\Lambda)\, \frac{d}{\sigma_i} \left(\frac{\|\mathbf{v}-\mathbf{w}_i\|^2}{d\,\sigma_i^2} - 1 \right), \quad \forall i. \tag{26}$$

Apparently, these learning rules correspond to those of eqs. (11,12), except for the constant factor d in eq. (26), but this can be absorbed by the learning rate η_σ. Hence, at least for our incomplete gamma distribution kernels,

and our activity-based definition of "winner", joint entropy maximization and Kullback-Leibler divergence minimization seem to be equivalent. This is a potentially interesting observation for density estimation since the log-likelihood of the training samples will equal the Kullback-Leibler divergence, when the number of training samples goes to infinity, and when the sample generating process is ergodic (for references, see Yin and Allinson, 2001).

As an example, we consider the distribution shown in Fig. 5A (quadrimodal product distribution). We have used this example before as a benchmark for comparing the density estimation performance of a series of kernel-based and kernel-extended topographic map formation algorithms (chapter 5, Van Hulle, 2000). The analytic equation of the distribution is, in the first quadrant, $(-\log v_1)(-\log v_2)$, with $(v_1, v_2) \in [0,1]^2$, and so on. Each quadrant is chosen with equal probability. The resulting asymmetric distribution is unbounded and consists of four modes separated by sharp transitions (discontinuities), which makes it difficult to model. The support of the distribution is bounded by the unit square $[-1,1]^2$. We take a 24×24 lattice and train it until $t_{max} = 1,000,000$ in eq. (13). The density estimate is obtained by using eq. (20), with $\mathbf{w}_i$ and σ_i as determined with our learning algorithm. The result is shown in Fig. 5B. The Kullback-Leibler divergence is 19.2; the Mean Squared Error (MSE) between the estimated and the theoretical distribution is 5.08×10^{-2}. For comparison's sake, we have also considered the classic Variable Kernel (VK) density estimation method (Silverman, 1992), which puts a Gaussian kernel at each input sample, and which adapts the kernel range to the local sample density.[4] The Kullback-Leibler divergence is 80.4 (MSE $= 7.12 \times 10^{-2}$)

[4] Technically, we take for the sensitivity parameter of the VK method, $\alpha = \frac{1}{2}$, as suggested by Breiman and coworkers (Breiman *et al.*, 1977). The pilot estimate was determined by using the (kth) nearest-neighbor method with $k = \sqrt{M}$ (Silverman, 1992).

for $M = 500$ samples, 38.6 (MSE $= 5.92 \times 10^{-2}$) for $M = 2000$ samples, 28.6 (MSE $= 5.57 \times 10^{-2}$) for $M = 5000$ samples, and 21.4 (MSE $= 5.35 \times 10^{-2}$) for $M = 10,000$ samples. Hence, apparently many more kernels are needed to match our learning algorithm's performance. Finally, in order to see the effect of our activity-based WTA competition, $i^* = \arg\max_{\forall i \in A} y_i$, we have also trained our lattice using the classic distance-based rule, $i^* = \arg\min_i \|\mathbf{w}_i - \mathbf{v}\|$ (and with the radii adapted as in eq. (12)). The resulting density estimate is shown in Fig. 5C. The Kullback-Leibler divergence is now 84.7 (MSE $= 5.13 \times 10^{-2}$), which is again inferior to our initial result.

8 Correspondence with other kernel-based topographic map algorithms

The Soft Topographic Vector Quantization (STVQ) algorithm (Graepel *et al.*, 1997) performs a fuzzy assignment of input samples to lattice neurons, similar to fuzzy clustering. It also serves as a general model for probabilistic, SOM-based topographic map formation since several algorithms can be considered as special cases, including Kohonen's Batch Map version (Kohonen, 1995). Our learning algorithm is different in at least three ways. First, the STVQ kernel represents a *fuzzy membership (in clusters) function*, *i.e.*, the softmax function, normalized with respect to the other lattice neurons. In our case, the kernel represents a cumulative distribution function, which operates in the input space, and determines the winning neuron. Second, instead of using kernels with equal radii in the STVQ algorithm, our radii are individually adapted. Third, the kernels also differ conceptually since in the STVQ algorithm, the kernel radii are related to the magnitude of the noise-induced change in the cluster assignment (thus, in lattice space), whereas in our case they are related to the radii of the incomplete gamma distribution kernels and, by consequence, to the standard deviations of the assumed Gaussian local input densities (thus,

in input space).

In the Kernel-based Soft Topographic Mapping (STMK) algorithm (Graepel *et al.*, 1998), a non-linear transformation, from the original input space to some "feature" space, is introduced that admits a kernel function, *e.g.*, a Gaussian. The topographic map's parameters ("weights") are expressed as linear combinations of the transformed inputs, so that the map is in effect developed in the feature space, rather than in the input space directly, as in our approach. Other points of distinction are that the kernels' parameters are not updated in the STMK algorithm, and that the inputs are assigned in probability to neurons. In András' approach (2001), the Gaussians serve as a non-linear transformation, mapping the input vectors to a high-dimensional space, so that the boundaries between the input classes are linearized. The kernel radii are adapted individually so that the map's classification performance is optimized, an operation that requires supervised learning. This basically sets András' approach apart from ours.

In the kernel-based Maximum Entropy learning Rule (kMER) (Van Hulle, 1998), the kernel outputs are thresholded (0/1 activations) and, depending on these binary activations, the kernel centers and radii are adapted. In the current approach, both the activation states as well as the learning rules eqs. (11,12) depend on the continuously graded kernel outputs.

In the approach of Benaim and Tomasini (1991), the weights are adapted in such a manner that the Kullback-Leibler divergence is minimized, given a *homoscedastic* Gaussian mixture density model. Furthermore, in order to achieve a topology-preserving mapping, a smoothness term of the form $\Lambda(i, i^*)(\mathbf{w}_i - \mathbf{w}_{i^*})$ is *added* to the weight update rule, with the winning neuron i^* defined by a distance-based WTA rule. Recently, Yin and Allinson (2001) re-considered Benaim and Tomasini's idea, and extended it by also adapting

the radii of the Gaussian kernels. Yin and Allinson's neighborhood function basically corresponds to the $\widehat{P}(\mathbf{v}|i)$ term in eqs. (23,24), which is in input space coordinates, instead of in lattice space coordinates, as in our case (*cf.*, $\Lambda(.)$ in eqs. (25,26)). Furthermore, the winning neuron i^* is the one for which the Gaussian kernel output is maximal. In general, this leads to quite different results.

9 Conclusion

We have developed a new learning algorithm for kernel-based topographic map formation. The algorithm is aimed at maximizing the joint entropy of the map's output. We have formulated the theoretically optimal joint entropy performance for two example input distributions, which we used for assessing the algorithm's performance, and also for suggesting a new type of clustering algorithm. Finally, we have shown the correspondence with stochastic gradient descent on the Kullback-Leibler divergence, in the case of a heteroscedastic Gaussian mixtures density model.

Acknowledgments

M.M.V.H. is supported by research grants received from the Fund for Scientific Research (G.0185.96N), the National Lottery (Belgium) (9.0185.96), the Flemish Regional Ministry of Education (Belgium) (GOA 95/99-06; 2000/11), the Flemish Ministry for Science and Technology (VIS/98/012), and the European Commission (QLG3-CT-2000-30161 and IST-2001-32114).

References

András, P. (2001). Kernel-Kohonen networks. *Int. J. Neural Systems*, submitted.

Bell A.J., Sejnowski, T.J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computat.*, **7**, 1129–1159.

Benaim, M., and Tomasini, L. (1991). Competitive and self-organizing algorithms based on the minimization of an information criterion. *Proc. ICANN'91*, pp. 391-396.

Bishop, C.M., Svensén, M., and Williams, C.K.I. (1998). GTM: The generative topographic mapping. *Neural Computat.*, **10**, 215-234.

Breiman, L., Meisel, W., and Purcell, E. (1977). Variable kernel estimates of multivariate densities. *Technometrics*, **19**, 135-144.

Durbin, R., and Willshaw, D. (1987). An analogue approach to the travelling salesman problem using an elastic net method. *Nature*, **326**, 689-691.

Gordon, A. (1999). *Classification.* (2nd edition) London: Chapman and Hall/CRC Press.

Graepel, T., Burger, M., and Obermayer, K. (1997). Phase transitions in stochastic self-organizing maps. *Physical Rev. E*, **56**(4), 3876-3890.

Graepel, T., Burger, M., and Obermayer, K. (1998). Self-organizing maps: Generalizations and new optimization techniques. *Neurocomputing*, **21**, 173-190.

Graham, R.L., Knuth, D.E., and Patashnik, O. (1994). *Concrete Mathematics: A Foundation for Computer Science.* Reading, MA: Addison-Wesley.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biol. Cybern.*, **43**, 59-69.

Kohonen, T. (1995). *Self-organizing maps.* Heidelberg: Springer.

Kostiainen, T., and Lampinen, J. (2002). Generative probability density model in the self-organizing map. In *Self-organizing neural networks: Recent advances and applications*, U. Seiffert & L. Jain (Eds.), pp. 75-94. Physica Verlag.

Linsker, R. (1989). How to generate ordered maps by maximizing the mutual information between input and output signals. *Neural Computat.*, **1**, 402-411.

Robbins, H., and Munro, S. (1951). A stochastic approximation method. *Ann. Math. Statist.*, **22**, 400-407.

Silverman, B.W. (1992). *Density estimation for statistics and data analysis.* London: Chapman and Hall.

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

Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a dataset via the Gap statistic. *J. Royal Statist. Soc. B*, **63**, 411-423.

Utsugi, A. (1997). Hyperparameter selection for self-organizing maps. *Neural Computat.*, **9**, 623-635.

Van Hulle, M.M. (1998). Kernel-based equiprobabilistic topographic map formation. *Neural Computat.*, **10**(7), 1847-1871.

Van Hulle, M.M. (2000). *Faithful representations and topographic maps: From distortion- to information-based self-organization.* New York: Wiley.

Vapnik, V.N. (1995). *The nature of statistical learning theory.* New York: Springer-Verlag.

Weisstein, E.W. (1999). *CRC Concise Encyclopedia of Mathematics.* London: Chapman and Hall.

Yin, H., and Allinson, N.M. (2001). Self-organizing mixture networks for probability density estimation. *IEEE Trans. Neural Networks*, **12**, 405-411.

Appendix

Derivation of learning rules eqs. (7,8)

We first consider the update rule for the kernel centers. The first term on the right hand side of eq. (6) does not depend on the kernel center. Hence, we only need to further concentrate on the second term, which in fact corresponds to the expected value of its ln component. Entropy maximization can be achieved by considering the training set of r's to approximate the density $p_r(r)$, which leads to the "on-line" stochastic gradient ascent learning rule:

$$\Delta \mathbf{w}_i = \eta_w \frac{\partial H}{\partial r}\frac{\partial r}{\partial \mathbf{w}_i} = \eta_w \left(\frac{\partial y_i}{\partial r}\right)^{-1} \frac{\partial}{\partial \mathbf{w}_i}\left(\frac{\partial y_i}{\partial r}\right), \quad \forall i. \tag{27}$$

After some algebraic manipulations, we obtain:

$$\frac{\partial y_i}{\partial r} = \frac{-2}{\Gamma(\frac{d}{2})}\frac{r^{d-1}}{(\sqrt{2}\sigma_i)^{d-1}} \exp\left(-\frac{(\frac{r}{\sigma})^2}{2}\right), \tag{28}$$

of which also the derivative with respect to $\mathbf{w}_i$ is needed. For the one dimensional case ($d = 1$), this leads to:

$$\Delta w_i = \eta_w \frac{v - w_i}{\sigma_i^2}, \quad \forall i, \tag{29}$$

when using eq. (3). For $d > 1$, we obtain the same term on the right hand side, and a second one: $(d-1)\frac{\mathbf{v}-\mathbf{w}_i}{\|\mathbf{v}-\mathbf{w}_i\|^2}$. This more complex rule also converges towards the centroid of the inputs that activate neuron i. It is reasonable to omit this second term, in order to keep the learning rule simple. Indeed, we know that $E[\|\mathbf{v} - \mathbf{w}_i\|^2] = d\sigma_i^2$, for a Gaussian input distribution, when $\mathbf{w}_i$ converges to the Gaussian's mean and σ_i to its standard deviation. Hence, the second term is expected to be smaller than the first term. We can also motivate the omission in the following manner. Since a d-dimensional radially-symmetrical Gaussian distribution can be built up by taking d samples – one for each input dimension – of a 1-dimensional Gaussian with the same radius, when the updates Δw_{ij} are small, and when we update along each input dimension

separately, *e.g.*, in random order, then we can approximate the learning rule by the simpler one eq. (7).

The update rule for the kernel radii eq. (8) can be derived directly, without any approximations, by performing gradient ascent on the differential entropy H, after some algebraic manipulations.

Derivation of learning rules eqs. (23,24)

We first consider the derivation of the update rule for the kernel centers, eq. (23). Since the true input density $p(\mathbf{v})$ is not known, Robbins-Munro stochastic approximation can be invoked in order to solve eq. (21). This leads to:

$$\Delta \mathbf{w}_i \;=\; \eta_w \, \frac{1}{\widehat{p}(\mathbf{v}|\Theta)} \frac{\partial \widehat{p}(\mathbf{v}|\Theta)}{\partial \mathbf{w}_i}, \quad \forall i. \tag{30}$$

Working out the derivative $\frac{\partial \widehat{p}(\mathbf{v}|\Theta)}{\partial \mathbf{w}_i}$ yields:

$$\frac{\partial \widehat{p}(\mathbf{v}|\Theta)}{\partial \mathbf{w}_i} = p_i(\mathbf{v}|\theta_i) \frac{(\mathbf{v} - \mathbf{w}_i)}{\sigma_i^2}, \tag{31}$$

with $p_i(\mathbf{v}|\theta_i) \stackrel{\Delta}{=} \frac{1}{(2\pi)^{\frac{d}{2}} \sigma_i^d} \exp\left(-\frac{\|\mathbf{v}-\mathbf{w}_i\|^2}{2{\sigma_i}^2}\right)$ and $\theta_i = \{\mathbf{w}_i, \sigma_i\}$. Bayes' rule tells us that:

$$\widehat{P}(\mathbf{v}|i) = \frac{\widehat{P_i}\; p_i(\mathbf{v}|\theta_i)}{\widehat{p}(\mathbf{v}|\Theta)}, \tag{32}$$

with $\widehat{P_i}$ the prior probability, *i.e.*, the ith mixing parameter in our Gaussian mixture. Now since we have assumed equal mixings in eq. (20), $\widehat{P_i}$ is constant. Substituting eqs. (31,32) in eq. (30) leads to the end result given in eq. (23).

In a similar way, the update rule for the kernel radii eq. (24) is obtained.

Figure captions

Fig. 1: Distribution functions of the Euclidean distance r from the mean of a unit-variance, radially-symmetrical Gaussian, parameterized with respect to the dimensionality d (continuous lines), and the corresponding complements of the cumulative distribution functions (dashed lines). The functions are plotted for $d = 1, \ldots, 10$ (from left to right); the thick lines correspond to the $d = 1$ case.

Fig. 2: Evolution of a 24×24 lattice as a function of time. *Left column:* evolution of the kernel weights. *Right column:* evolution of the kernel radii. The boxes outline the uniform input probability density. The values given below them represent time.

Fig. 3: (A,B) Optimal joint entropy (JE – thick continuous lines) and mutual information (MI – thick dashed lines) for the case of N lattice neurons and N 1-dimensional Gaussians spaced infinitely far apart (A), and for the case of only one 1-dimensional Gaussian (B). Results are plotted as a function of the number of neurons N, parameterized with respect to $k = 2, 4, 8, 16, 32$ quantization levels, with the $k = 2$ curves being the lowest and the $k = 32$ curves being the highest ones. The thin dashed line in (A) denotes the MI plot for the case where $k \to \infty$; the dot-dashed line the case where $k, N \to \infty$. The thin continuous line in (B) is the average MI result obtained with the learning algorithm (including error bars).
(C) Simulation results obtained for JE, for the N Gaussians case with $k = 2$, plotted as a function of the dimensionality d and parameterized with respect to the number of lattice neurons N, $N = 1, 2, \ldots, 10$. The thick line corresponds to $N = 1$ case, the upper line to $N = 10$.

(D) Optimal JE and MI (thick continuous and dashed lines, respectively) for the 1-Gaussian case, plotted as a function of d. Note that MI $= 0$ for $N = 1$. The thin continuous line is the average MI result (error bars omitted) obtained with our learning algorithm. Other conventions are as in Fig.3C.

Fig. 4: Clustering based on the difference between expected and obtained joint entropy (thick continuous line), and based on the Gap statistic applied to kernel weights (thin continuous line) and applied to the k-means clustering result (dashed line).

Fig. 5: Two-dimensional quadrimodal product distribution (A), and density estimate obtained with our kernel-based topographic map learning algorithm (B), and when in our learning algorithm a distance-based competition is used instead of an activity-based (C).

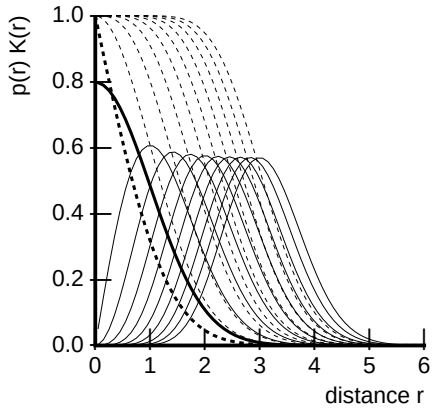

Figure 1:

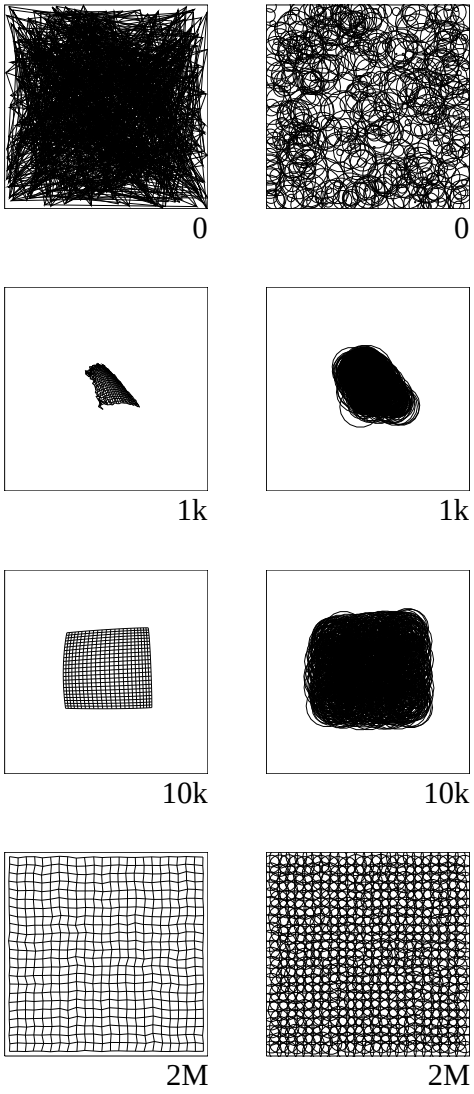

Figure 2:

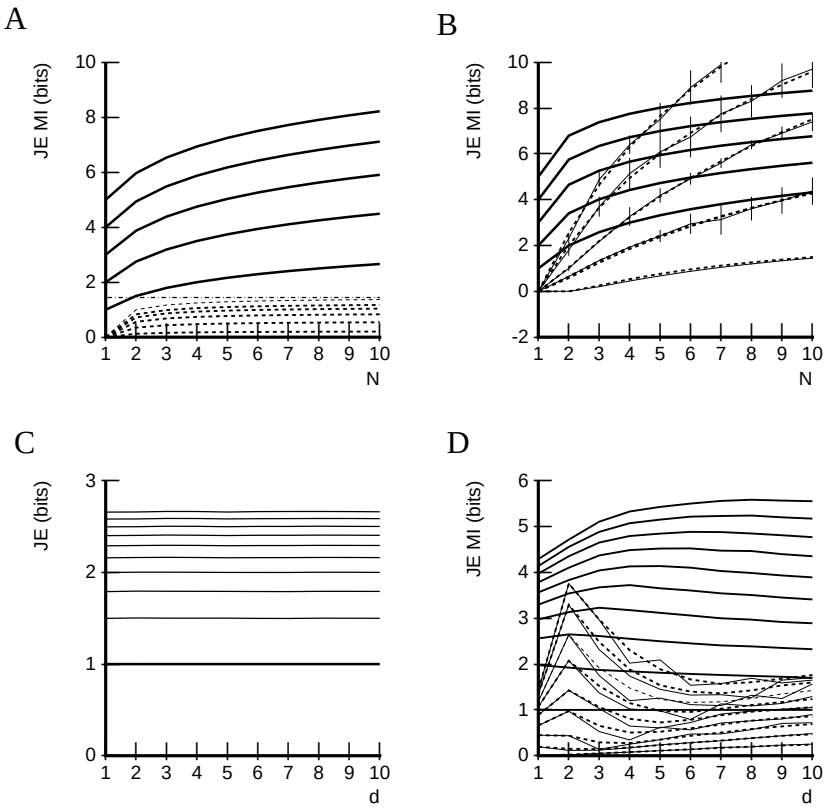

Figure 3:

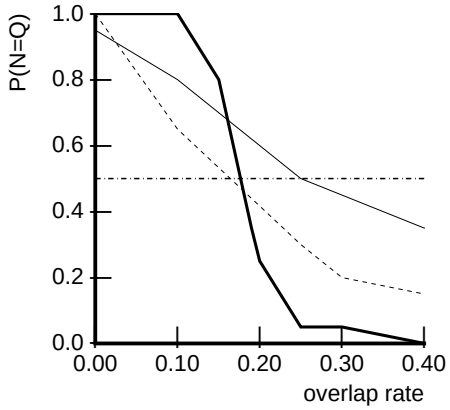

Figure 4:

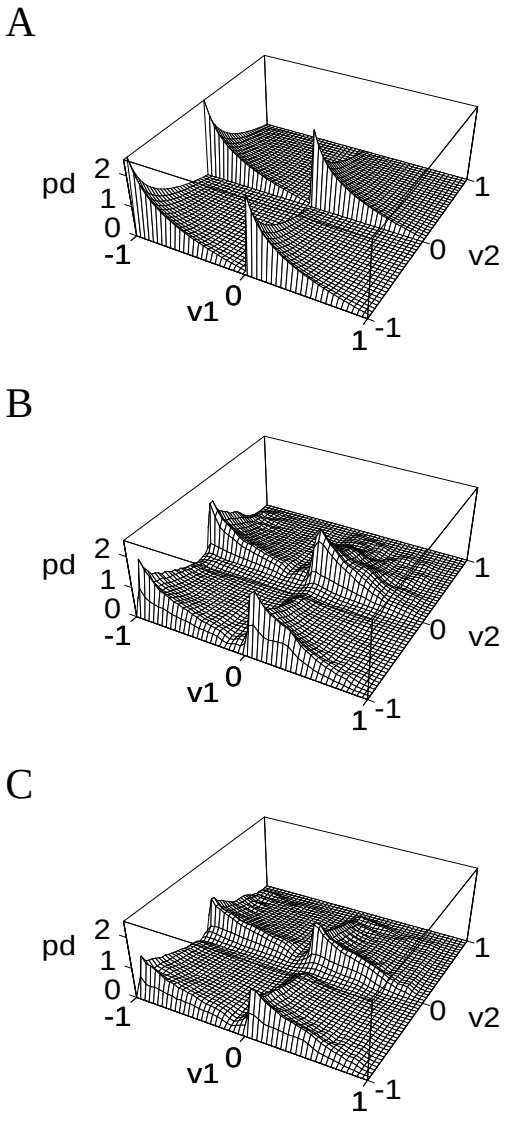

Figure 5: