

# On the Generative Probability Density Model in the Self-Organizing Map

Timo Kostiainen and Jouko Lampinen

*Laboratory of Computational Engineering, Helsinki University of Technology,  
P.O.Box 9400, FIN-02015 ESPOO, FINLAND  
Timo.Kostiainen@hut.fi, Jouko.Lampinen@hut.fi*

---

## Abstract

The Self-Organizing Map, SOM, is a widely used tool in exploratory data analysis. A major drawback of the SOM has been the lack of a theoretically justified criterion for model selection. Model complexity has a decisive effect on the reliability of visual data analysis, which is a main application of the SOM. In particular, independence of variables cannot be observed unless generalization of the model is good. We describe the maximum likelihood probability density model which follows from the SOM training rule, and show how the density model can be applied to choosing the correct model complexity, based on the method of maximum likelihood.

*Keywords:* self-organizing map, density model, maximum likelihood, model selection, data analysis

---

## 1 Introduction

In this paper we formulate the generative probability density model which stems from the SOM training rule [4]. The density model gives new insight to the interpretation of the SOM, which can help to assess the suitability of the SOM to a given practical problem. The density model also enables the application of tools of probability theory to choosing the correct complexity for the SOM model. There are various applications for the SOM algorithm, but we limit our attention to data mining, or the finding of statistically significant dependencies in a multidimensional data sample.

Perhaps the most valuable property of the SOM algorithm is the preservation of topology or the fact that neighborhood relationships of the input data are maintained in the mapping. For that reason, the issue of parameter selection has been focused on this concept. A number of different measures have been

developed, but none of them have gained much popularity among SOM users. Currently, there is no clear consensus on the precise definition of topology preservation. Although data analysis is a prominent application of the SOM, very few publications have treated the SOM as a statistical model. In data analysis, the algorithm is used to produce a model of the input data distribution. It is standard practice to visually examine a two-dimensional map embedded in high-dimensional data and to make inferences about the data based on this model. In this case, the obvious criterion for model selection should be generalization to new data, just as it is in the case of any other method for statistical modeling. The preservation of topology is also important but it is second to generalization, if the main concern is the statistical reliability of interpretations. However illustrative the visual display may appear to be, it is almost useless in the analysis of a sample from a population if such phenomena show up on the map which do not generalize to the whole population. Therefore it is essential that the generalization of the model can be measured.

Mathematical analysis of the SOM has proved fairly difficult due to the heuristic origin of the algorithm. The algorithm is not defined in terms of an error function, but directly via the update rule. It has been shown by Erwin *et al* [2] that the training rule cannot be the gradient of any global error function. The algorithm does converge to the local minimum of an error function [7], but during training the function keeps changing and the function does not exist if the input distribution is assumed continuous. However, it has been pointed out by Luttrell [6] that a slight change in the winner selection rule, which does not appear to significantly alter the properties of the SOM, results in a continuous global error function. Luttrell refers to this as the minimum distortion rule. Alternatives to the SOM have been developed in order to overcome the theoretical problems and to enable probabilistic analysis. Examples of these include the Generative Topographic Mapping [1] and the approach taken by Utsugi [8]. Both of these approaches explicitly include a generative density model, which is a constrained Gaussian mixture model. Model selection can then be based on maximum likelihood or Bayesian evidence or some other well defined criterion.

We formulate the probability density model for which SOM training gives the maximum likelihood estimate, based on the local error function, and show a method for computing the values of the density function. We discuss the properties of the density model, which are directly linked to the properties of the SOM model itself. We believe that this aspect may be beneficial in further research into the unique properties of the SOM, including the topological behavior. The density model also bridges the gap between the SOM and other statistical modeling techniques and thus helps in choosing between them. The density model is needed to compute the likelihood of data for a SOM model with a given set of parameters. The likelihood of data is a widely accepted

criterion for model selection.

In the case of the original SOM algorithm, the density model exists only after training has converged to a local minimum of the error function. While this is quite adequate for the purposes of data analysis, the theoretical problems can be avoided by adopting the minimum distortion rule.

## 2 Probability density model in the self-organizing map

We consider the probability density model for input data which is implied by the SOM algorithm. In the interpretation of Luttrell [5], the neighborhood function of the SOM can be thought of as the noise model of the internal representation of the input data (the encoded state). What we consider is the noise model or probability density model for the input data  $p(x)$  which follows from the internal noise model of the neighborhood. We do not make any assumption of the shape of the neighborhood function (except in numerical examples).

The converged state of the SOM is a local minimum of the error function which is given by [7]

$$E = \sum_{n=1}^N \sum_{r=1}^M H(b(x^n) - r) \|x^n - m^r\|^2, \quad (1)$$

where  $X = \{x^n\}, n = 1, \dots, N$  is the discrete data sample,  $r$  is the index (or position) of a unit in the SOM,  $b(x)$  is the index of the best matching unit for  $x$ ,  $m^r$  is the reference vector of the unit  $r$ , and  $H()$  is the neighborhood function. The error function is only defined for a discrete data set and fixed neighborhood parameters, and as such it cannot be used to compare maps which have different neighborhoods.

The method of maximum likelihood (ML) means choosing that model  $m$  which has the highest likelihood of having generated the input data  $X$ . If the data samples  $x^n$  are assumed independent, the likelihood  $p(X|m)$  is the product of sample probabilities  $\prod_n p(x^n|m)$ . Maximizing the likelihood function is equivalent to minimizing its negative logarithm  $-\log(p(X|m)) = -\sum \log(p(x^n|m))$ . If the noise model of the samples  $p(x^n|m)$  is Gaussian, the negative log-likelihood is the sum of square errors. We start from the square error function eq. (1), and we wish to find a likelihood function that is consistent with this error function. Let  $p(X|m, H) = \prod_n p(x^n|m, H)$  be the likelihood of the input data  $X$  given the SOM model with codebook  $m$  and neighborhood function  $H$ . Then  $p(x|m, H)$  represents the probability density function of the input data, which we want to solve. The negative log-likelihood is  $L = -\log p(X|m, H)$

and setting it proportional to eq. (1) yields

$$\begin{aligned} p(X|m, H) &= Z' \exp(-\beta E) = Z' \exp(-\beta \sum_n \sum_r H(b(x^n) - r) \|x^n - m^r\|^2) \\ &= Z' \prod_n \exp(-\beta \sum_r H(b(x^n) - r) \|x^n - m^r\|^2). \end{aligned} \quad (2)$$

Here we have introduced two constants,  $Z'$  and  $\beta$ , which are not needed in the ML estimate of the codebook  $m$  but which are necessary for the complete density model. The likelihood of data is the product of sample probabilities, so the probability density function is given by

$$p(x|m, H) = Z \exp(-\beta \sum_r H(b(x) - r) \|x - m^r\|^2). \quad (3)$$

From this form of the density function it appears that the density at any point is the product of a number of Gaussian density components. Note that the discontinuity of the function at the boundaries of Voronoi cells is due to the discontinuity of the best-matching unit selection rule of the SOM algorithm. Another way to write the same function is to complete the sum (over  $r$ ) of squares into a square-of-sum form. Algebraic manipulations yield the following form:

$$p(x|x \in V_b) = Z e^{-\beta W_b} \exp\left(-\frac{1}{2s_b^2} \|x - \mu_b\|^2\right), \quad (4)$$

where, adopting the notation  $H_{bj} = H(b(x) - j)$ ,

$$\mu_b = \frac{\sum H_{bj} m_j}{\sum H_{bj}} \quad (5)$$

$$s_b^2 = 1/(2\beta \sum H_{bj}) \quad (6)$$

$$W_b = \sum H_{bj} \|m_j - \mu_b\|^2, \quad (7)$$

and  $V_b$  is the Voronoi cell or receptive region associated with the reference vector  $m_b$ . From eq. 4 it is obvious that the density function has Gaussian form within each Voronoi cell. At cell boundaries, the density may be discontinuous. See Figs. 1 and 2 for examples of the density model.

The normalizing constant  $Z$  and the noise variance parameter  $\beta$  are bound together by the constraint that the integral of the density over the data space must equal one. That integral can be written as

$$\int p(x) dx = Z \sum_r e^{-\beta W_r} \int_{x \in V_r} \exp\left(-\frac{1}{2s_r^2} \|x - \mu_r\|^2\right) dx, \quad (8)$$

where the integration over  $x$  is decomposed to the sum of integrals over each Voronoi cell. The integrals cannot be computed in closed form but they can

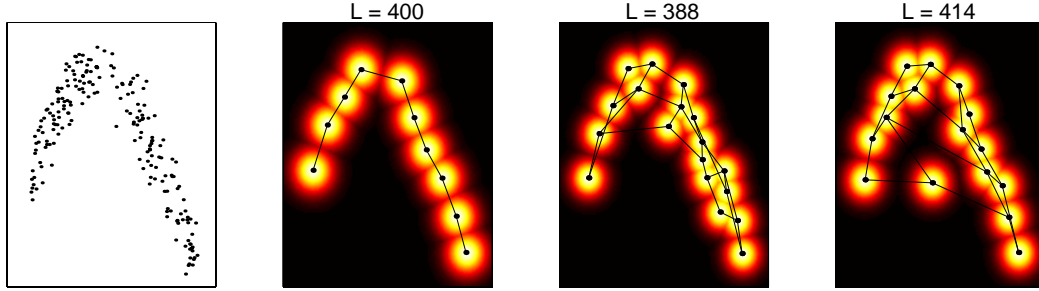


Fig. 1. Training data and density estimates due to different SOM topologies.  $L$  denotes the negative log-likelihood of test data. Between these options, the  $3 \times 6$  topology (middle) produces the best model judging by the ML criterion.

be approximated numerically using Monte Carlo sampling. A simple way to do this is the following algorithm:

- (1) For each cell  $r$ , draw  $L$  samples from the normal distribution  $N(\mu_r, s_r)$
- (2) Compute  $q_r = L_r/L$ , the fraction of samples that are inside the cell  $r$ .
- (3) The integral over  $V_r$  in eq. (8) equals  $q_r(2\pi s_r^2)^{d/2}$ , where  $d$  is the dimension of the data space.

For a map that contains  $M$  units, this algorithm requires the computation of distances between  $M \times L$  samples and the  $M$  reference vectors. Thus if  $M$  is large the computational cost by far exceeds that of the training algorithm itself. In an efficient implementation the number of samples  $L$  should be chosen according to the desired accuracy. The acceptance ratio  $q_r$  varies very much depending on the situation. When the neighborhood is small, the neighborhood-weighted center  $\mu_r$  is close to the reference vector  $m_r$  and  $s_r$  is likely to be small, so  $q_r$  is high. When the neighborhood is large the situation is opposite and to achieve an equivalent accuracy  $L$  will have to be much greater. Detailed analysis of the dependence of the accuracy on  $L$  is presented in appendix A.

Above, we described a technique for determining  $Z$  for a given value of  $\beta$ . For the ML estimate of the density model, one needs the value of  $\beta$  that maximizes the likelihood function (2). This can only be found by numerical search by computing the likelihood of data for different values of  $\beta$  (after solving  $Z$ ). It is worth noting that when an algorithm such as bisection search is applied, savings can be made by allowing the accuracy to vary. Initial estimates can be very coarse, corresponding to small  $L$ , if the accuracy gradually increases towards the convergence of the search. The final accuracy should reflect the size of the validation data sample.

By equating the partial derivative of the likelihood function  $\partial p(X)/\partial \beta$  with zero, an interpretation for the maximum likelihood solution  $\beta^{\text{ML}}$  can be found

in terms of the neighborhood-weighted distortion function

$$D(x) = \sum_j H_{bj} \|x - m_j\|^2, \quad (9)$$

as follows:

$$\frac{\sum_i^N D(x_i)}{N} = \frac{\int D(x) \exp(-\beta^{\text{ML}} D(x)) dx}{\int \exp(-\beta^{\text{ML}} D(x)) dx}. \quad (10)$$

Observe that the estimated input distribution is  $\hat{p}(x|\beta, H) \propto \exp(-\beta D(x))$ . Heuristically, eq. (10) says that at the ML-estimate  $\beta = \beta^{\text{ML}}$  the mean value of  $D(x)$  over the estimated input distribution equals the sample average of  $D(x_i)$  over the input data  $x_i, i = 1, \dots, N$ .

The difference between the SOM density model and the usual additive Gaussian mixture model is in the hard assignments to nearest density kernels. The softening effect of the neighborhood is in the internal representation of the SOM, i.e. the topological index space. The contribution of nearby units to the density does not depend so much on the distance between the point and these units' reference vectors, as it does on the topological relationship between these units and the best matching unit. The fact that the density is the product of Gaussian components is due to the square-error distortion measure, which is a result of the nearest-neighbor winner selection rule of the SOM. The form of the density model is a direct consequence of the SOM training rule which minimizes the error function (1). Hence, the discontinuity and the fact that the density kernels in eq. (4),  $\mu_j$ , do not coincide with the reference vectors,  $m_j$ , are properties of the SOM algorithm and something that should be considered when deciding whether the SOM is the right solution to a given practical task.

### 3 Model selection

The SOM algorithm produces a model of the input data. The complexity of this model is determined by the number of units and the width of the neighborhood, which has a regularizing effect on the model. When the input data is a sample from a larger population, the objective is to choose the complexity such that the model generalizes as well as possible to new samples from that population. (The situation is different if the data comprises the entire population and the effect of measurement noise is considered negligible. In that case, each sample is significant and overfitting will not be an issue.)

Let us first regard the number of units as given, so neighborhood width  $\sigma$  is the sole control parameter. The density model allows us to select the neighborhood width  $\sigma$  by maximizing the likelihood of data  $p(X|m, H)$ . In the

course of SOM training,  $\sigma$  is gradually decreased in some pre-specified manner, i.e.  $\sigma = \sigma(t), t = 1, \dots, K; \sigma(t+1) < \sigma(t)$ . We trust that the training algorithm will find an ML estimate for the map codebook at each value of the neighborhood width  $\sigma(t)$ , if it is allowed to converge every time. To construct the density model for each of these  $K$  candidate maps, we optimize  $\beta(t)$  numerically as described in the previous section. This yields  $K$  different density models to compare. To choose between these we compute the likelihood values  $p(X_V|m(t), \sigma(t), \beta^{\text{ML}}(t))$  for validation data  $X_V$  (which ideally should be different from that which was used to select  $\beta^{\text{ML}}(t)$ ). Cross-validation can also be applied. An example of model selection is shown in Fig. 2. The third map from the left maximizes the likelihood of validation data. This approach

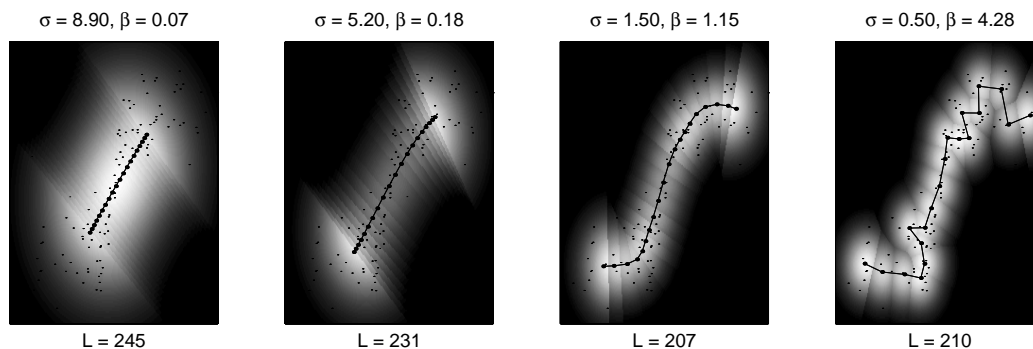


Fig. 2. SOM density models for different widths  $\sigma$  of the Gaussian neighborhood. From the total likelihood of validation data the optimal neighborhood can be chosen to avoid overfitting.  $L$  denotes the negative log-likelihood of validation data. The samples were taken from the tanh function.

extends directly to the comparison of different size maps as well as different topologies (see Fig. 1). If one wishes to have a large map, it may be advisable to ease the computational requirement by finding the correct  $\sigma$  for a smaller map first and then simply scaling it up in proportion to the dimensions of the maps. (For example, if  $\sigma_{KL}$  is the optimal neighborhood width for a  $K \times L$  map, then  $5\sigma_{KL}$  is probably a reasonable value for a  $5K \times 5L$  map.)

Because the exact value of the density function cannot be computed in closed form, it is difficult to apply methods such as Bayesian evidence to parameter selection. If the values of the function itself are approximations, then the derivatives will be even more inaccurate.

A common application of the SOM is to look for dependencies between variables by visual inspection. In that context, the density model can be used to select the complexity of the model, but it also enables quantitative analysis. Regression or conditional expectations can be computed directly from the joint density (3) by numerical integration. For example, the conditional

distribution for variable  $x_j$  equals

$$p(x_j|x_{\setminus j}, m, H) = \frac{p(x|m, H)}{\int p(x|m, H) dx_j}, \quad (11)$$

where  $x_{\setminus j}$  denotes the vector  $x$  with element  $j$  excluded. Likewise, the regression of  $x_j$  on other variables can be computed as the conditional mean  $E[x_j|x_{\setminus j}, m, H]$ . It should be noted that the SOM density model may not give the best possible description of the input distribution. We have included this discussion here so as to illustrate the value of model selection.

Reducing the variance parameter to zero,  $\beta \rightarrow \infty$ , gives an important special case. The conditional density is then sharply peaked at the value of the “outputs”  $x_j$  in the best matching unit for the “inputs”  $x_{\setminus j}$ . The conditional mean  $E[x_j|x_{\setminus j}]$  then gives the same value as nearest neighbor (NN) regression with the SOM reference vectors, with the neighborhood-weighted reference vectors (eq. 5) as output values, producing a piecewise constant estimate. Comparison with the NN rule is interesting, because it is a close quantitative counterpart of the popular visual analysis of the SOM.

Fig. 3 illustrates the difference between computing the conditional mean from the density model and using the nearest neighbor rule. A random 3D data set ( $\sim N(0, 1)$ ) is analyzed by a  $6 \times 6$  SOM. We attempt to infer  $E(x_2|x_1, x_3 = 0)$ , the expected value of the variable  $x_2$  given  $x_1$ , when  $x_3$  equals zero. As the variables are really independent, the answer should be  $E(x_2|x_1, x_3 = 0) = E(x_2) = 0$ . The optimal width of the Gaussian neighborhood function is  $\sigma = 4.2$ , which is a relatively large value, suggesting independent variables (a “simple” distribution). At zero neighborhood, the model is badly overfitted, and clearly neglecting to select the correct model complexity would give unreliable results. When the complexity is right, the nearest neighbor rule can give a good approximation to the mean and consequently the visual display will be as reliable as it can be. Confidence intervals computed from the density model can give further information of the reliability of data analysis.

By mere visual inspection of the map it is difficult to perceive the mean or shape of the conditional distributions and thus the reliability of the conclusions is very difficult to assess without a consistent measure of generalization, such as the likelihood of validation data. The examples shown in Figs. 2 and 3 indicate that this method will outperform any prefixed heuristic rule. However, it is not guaranteed that choosing parameter values to optimize the density estimate will result in a mapping that is also optimal for visual display, so it is always a good idea to validate the results of visual inspection by other, more reliable techniques. Yet a great amount of work will be avoided, if the most reliable map can be picked automatically from a group of candidates.

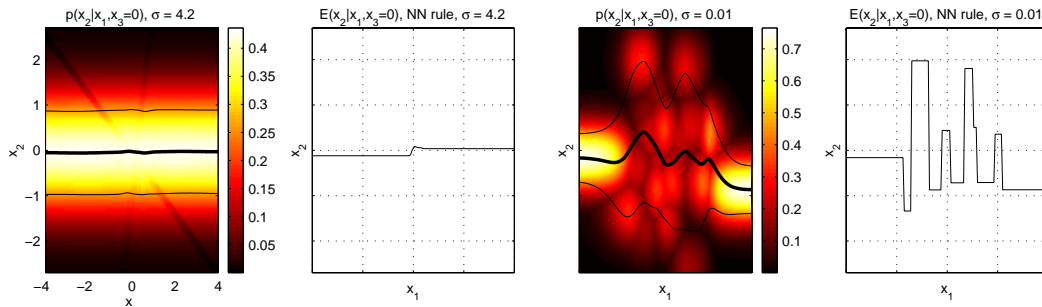


Fig. 3. Conditional densities from a SOM trained on random independent data. On the left the conditional density and the nearest neighbor prediction for optimal neighborhood  $\sigma = 4.2$ , and on the right for small neighborhood  $\sigma = 0.01$ , respectively. The curves represent means and standard deviations computed from the densities.

#### 4 Notes on the error function

In general, minimization of the error function eq. (1) does not lead to Kohonen's SOM training rule. Moreover, the error function is neither defined nor continuous at Voronoi cell boundaries. For this reason the error function only exists for a discrete data set, when the probability of any sample lying at any boundary is zero. The same applies to the density model. In practical data analysis we are always dealing with a discrete data set and the algorithm is allowed to converge whereby eq. (1) becomes valid, so the question of existence of the density is not a practical issue. The discontinuity of the density model is arguably a problem, which is directly due to the SOM training algorithm. The discontinuity may not show up as conspicuously in data analysis if the density model is not computed, but the same effects will be present anyway.

Luttrell [6] has shown that exact minimization of eq. (1) leads to the following approximation to the original training rule: Instead of the nearest neighbor winner rule,

$$b(x) = \underset{i}{\operatorname{argmin}} \|x - m_i\|, \quad (12)$$

the best matching unit is taken to be that which minimizes the value of the distortion function (9), i.e.

$$b(x) = \underset{i}{\operatorname{argmin}} \sum_j H_{ij} \|x - m_j\|^2. \quad (13)$$

The minimum distortion rule avoids many theoretical problems associated with the original rule, without compromising any desirable properties of the SOM (except for an increase in computational burden)[3]. The gradient of the error function becomes continuous at the boundaries of receptive fields (which are no longer the same as the Voronoi tessellation). The density model becomes continuous and differentiable, too.

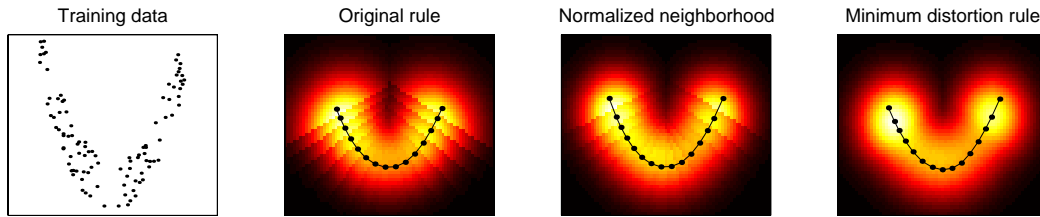


Fig. 4. Effect of the minimum distortion training rule on the density model. The neighborhood width is the same  $\sigma = 2.8$  for all the maps. (This value is larger than would be optimal, as the purpose of the figure is to highlight the differences between these cases.)

In typical implementations of the SOM, the neighborhood function is the same for each map unit. This causes problems near the borders of the map, where the neighborhood function gets clipped and thus becomes asymmetric. As a result, units close to the border get pulled excessively toward the center of the map, and the map does not extend close to the edges of the input distribution until the neighborhood is relatively small and the regularization is loose. This effect can be alleviated by normalizing the neighborhood function at the edges of the map. The portion of the neighborhood function that gets clipped off due to the finite size of the map lattice is transferred to nearest edge units. This is of particular importance, if the minimum distortion rule (13) is applied to winner selection. We see from eq. (6) that when the sum of the neighborhood function is constant throughout the map, all cells have equal noise variance.

In practice we find that all these variations will produce very similar results in terms of model selection. As can be seen in Fig. 4, neighborhood normalization smoothes out the density model somewhat. Note also the increased separation of the edge units from the rest. Obviously, the continuous density model is favorable in many respects.

## 5 Conclusions

We have discussed the difficulties that arise in the application of the SOM to statistical data analysis. We have shown how the probability density model can be used to find the maximum likelihood estimate for the SOM parameters, in order to optimize the generalization of the model to new data. The parameter search involves a considerable increase in the computational cost, but in our opinion it is outweighed by the benefit in reducing the number of false findings in data analysis. The model which gives the highest likelihood on validation data will most likely be the one that produces the most reliable visualization.

It should be stressed that although the density model is based on the error function which is not defined in all cases, in practice the only restriction in the

application of the density function to data analysis is that the algorithm should be allowed to converge. Unfortunately, maximizing the likelihood of data is not directly related with ensuring the reliability of the visual representation of the SOM. Especially when the data dimension is high, the conditional densities may easily end up being distributed around the map. In other words, the responsibility of a narrow range of values of some variable may be shared among many map units. Such effects cannot be observed by visual inspection of the component levels, and any hypotheses may need to be validated by other means. Confidence intervals which can be obtained from the density model may help to assess the quality of the visual display.

The association of a generative probability density model with the SOM enables comparisons between the SOM and the Generative Topographic Mapping, for example. If the theoretical difficulties of the SOM are avoided by adopting the minimum distortion winner selection rule, the main difference that remains is the hard vs. soft assignment of data to the units. The hard assignments of the SOM are perhaps easier to interpret and visualize, while the Gaussian mixture model may in many cases be more plausible. Apparently the choice of methods depends on the application.

Though this contribution concentrates on statistical data analysis it is likely that the approach presented here will be beneficial in some other applications of the SOM, as well. <sup>1</sup>

## References

- [1] Christopher M. Bishop, Markus Svensén, and Christopher K. I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10:215–234, 1998.
- [2] Ed Erwin, Klaus Obermayer, and Klaus Schulten. Self-organizing maps: Ordering, convergence properties and energy functions. *Biol. Cyber.*, 67(1):47–55, 1992.
- [3] Tom Heskes. Energy functions for self-organizing maps. In Erkki Oja and Samuel Kaski, editors, *Kohonen maps*, pages 303–315. Elsevier, 1999.
- [4] Teuvo Kohonen. *Self-Organizing Maps*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).
- [5] Stephen P. Luttrell. Derivation of a class of training algorithms. *IEEE Trans. on Neural Networks*, 1(2):229–232, June 1990.
- [6] Stephen P. Luttrell. Code vector density in topographic mappings: scalar case. *IEEE Trans. on Neural Networks*, 2(4):427–436, July 1991.

---

<sup>1</sup> Some MATLAB<sup>®</sup> code for evaluating the SOM density function is available at <http://www.lce.hut.fi/research/sompdf/>.

- [7] H. Ritter and K. Schulten. Kohonen self-organizing maps: exploring their computational capabilities. In *Proc. ICNN'88 Int. Conf. on Neural Networks*, volume I, pages 109–116, Piscataway, NJ, 1988. IEEE Service Center.
- [8] Akio Utsugi. Hyperparameter selection for self-organizing maps. *Neural Computation*, 9(3):623–635, 1997.

## A APPENDIX: Accuracy vs. number of MC samples

In the algorithm described in Section 2 one picks  $L$  samples and computes  $S$ , the number of samples that satisfy a certain condition. The task is to estimate  $q$ , the probability of a single sample satisfying the condition, based on  $S$  and  $L$ .  $S$  follows the binomial distribution

$$p(S|q, L) = q^S(1 - q)^{L-S} \quad (\text{A.1})$$

In Bayesian terms, the posterior distribution of  $q$  for given  $S$  and  $L$  is

$$p(q|S, L) = \frac{p(S|q, L)p(q)}{\int p(S|q, L)p(q) dq} = \frac{q^S(1 - q)^{L-S}}{\int q^S(1 - q)^{L-S} dq}, \quad (\text{A.2})$$

where the prior distribution  $p(q)$  is uniform. The integral in the denominator yields

$$\int q^S(1 - q)^{L-S} dq = \sum_{k=0}^{L-S} \frac{(-1)^k}{S + k + 1} \binom{L - S}{k}. \quad (\text{A.3})$$

Moments of the posterior can be written as serial expressions, for example

$$E(q) = \int qp(q|S, L) dq = \frac{\sum_{k=0}^{L-S} \frac{(-1)^k}{S+k+2} \binom{L-S}{k}}{\sum_{k=0}^{L-S} \frac{(-1)^k}{S+k+1} \binom{L-S}{k}}, \quad (\text{A.4})$$

and from the first and second moments we can derive an expression for the variance of the estimate  $\hat{q} = E(q)$ . Unfortunately, the computation easily runs into numerical difficulties due to the alternating sign  $(-1)^k$ . Exact values of the binomial coefficients are required, and these can be difficult to obtain if, say,  $L > 100$ .

To consider an approximation to the variance, observe that when  $S = L$  the formulas simplify considerably and the variance can be written as

$$\nu_0 = \text{Var}(\hat{q}|S = \{0, L\}) = \frac{L + 1}{(L + 3)(L + 2)^2}. \quad (\text{A.5})$$

This is the minimum value of the variance for given  $L$ . The variance of the *binomial distribution* at the maximum likelihood estimate  $q^{\text{ML}} = S/L$  equals  $q^{\text{ML}}(1 - q^{\text{ML}})/L$ . We can combine these results to get a fairly good approximation

$$\text{Var}(\hat{q}) \approx \nu^*(\hat{q}, L) = \nu_0 + \hat{q}(1 - \hat{q})/L, \quad (\text{A.6})$$

which slightly over-estimates the variance. A more precise approximation can be obtained directly from eq. (A.2) by numerical integration. Picking Monte Carlo samples from the distribution  $N(\hat{q}, \sqrt{\nu^*(\hat{q})})$  truncated to the range  $[0, 1]$  produces good results, except maybe at the very edges of the range, but there the exact value  $\nu_0$  can be applied.

Let us write the sum in eq. (8) as  $F = \sum w_i q_i$ , where  $w_i = e^{(-\beta W_b)} (2\pi s_i^2)^{(d/2)}$ . The relative standard error of an estimate of  $F$  equals

$$\epsilon = \frac{\Delta \hat{F}}{\hat{F}} = \frac{\sum w_i \Delta \hat{q}_i}{\sum w_i \hat{q}_i} = \frac{\sum w_i \sqrt{\text{Var}(\hat{q}_i)}}{\sum w_i \hat{q}_i} \quad (\text{A.7})$$

Hence, to achieve a given accuracy  $\epsilon$ ,  $L$  should be increased until

$$\frac{\sum w_i \sqrt{\text{Var}(\hat{q}_i)}}{\sum w_i \hat{q}_i} < \epsilon. \quad (\text{A.8})$$