

# Hyperparameter Selection for Self-Organizing Maps

Akio Utsugi

National Institute of Bioscience and Human-Technology,  
1-1 Higashi Tsukuba Ibaraki 305, Japan

October 8, 1996

## Abstract

The self-organizing map (SOM) algorithm for finite data is derived as an approximate MAP estimation algorithm for a Gaussian mixture model with a Gaussian smoothing prior, which is equivalent to a generalized deformable model (GDM). For this model, objective criteria for selecting hyperparameters are obtained on the basis of empirical Bayesian estimation and cross-validation, which are representative model selection methods. The properties of these criteria are compared by simulation experiments. These experiments show that the cross-validation methods favor more complex structures than the expected log likelihood supports, which is a measure of compatibility between a model and data distribution. On the other hand, the empirical Bayesian methods have the opposite bias.

## 1 Introduction

Recently, several standard learning methods for neural networks are being reconstructed as an estimation algorithm of a stochastic model. Such statistical treatment of learning enables inference at a higher level than a simple parameter estimation level, such as the evaluation of model reliability and automatic model selection. For example, MacKay (1992) studied backpropagation learning in a unifying Bayesian framework and presented a selection method of hyperparameters and model structure.

Among many learning methods, the self-organizing map (SOM) (Kohonen, 1988, 1990) is unique from the viewpoint of data analysis, because of the ability to extract topological structure hidden in data. However, since this learning

was originally defined only at an algorithmic level, its development to higher-level inference is difficult.

Statistical models behaving in a similar manner as SOM are also studied. The elastic net (Durbin et al., 1989), which is one of the generalized deformable models (GDM) (Yuille, 1990), learns a topology-preserving map as maximum a posteriori (MAP) estimates for the parameters of a Bayesian stochastic model. Although this model was studied originally in the context of an optimization problem such as the traveling salesman problem, it can be used for smoothing data along a specified topology. However, this requires the determination of two hyperparameters: the size of noise and the smoothness of route. The selection of these hyperparameters was attempted by an empirical Bayesian method similar to that of MacKay for backpropagation learning (Utsugi, 1993).

In the present paper, first, the above-mentioned hyperparameter selection method is reviewed. Next, the SOM algorithm for finite data is derived as an approximate MAP estimation algorithm for GDM. Thus, SOM and GDM can be considered equivalent at a stochastic model level. Finally, cross-validation, which is another representative model selection method, is applied to hyperparameter selection for our model and compared with the empirical Bayesian method by simulation experiments.

## 2 Empirical Bayesian hyperparameter selection for GDM

### 2.1 Construction of GDM as stochastic model

Initially, we regard the simplest type of *Gaussian mixture model* as a stochastic model for competitive learning (Nowlan, 1990). For a data set  $\mathbf{X}$  consisting of  $m$ -dimensional data points  $\mathbf{x}_i = (x_{i1}, \dots, x_{im})'$  ( $i = 1, \dots, n$ ), the likelihood function of the model with  $r$  components is given by

$$f(\mathbf{X}|\mathbf{w}, \beta) = \prod_{i=1}^n \sum_{s=1}^r \frac{1}{r} f(\mathbf{x}_i|\mathbf{w}_s, \beta) \quad (2.1)$$

where

$$f(\mathbf{x}_i|\mathbf{w}_s, \beta) = \left(\frac{\beta}{2\pi}\right)^{m/2} \exp\left(-\frac{\beta}{2}\|\mathbf{x}_i - \mathbf{w}_s\|^2\right) \quad (2.2)$$

is a component Gaussian density with the centroid  $\mathbf{w}_s = (w_{s1}, \dots, w_{sm})'$  and the variance  $1/\beta$ , and  $\mathbf{w} = (\mathbf{w}'_1, \dots, \mathbf{w}'_r)'$ .

Furthermore, we consider a *Gaussian smoothing prior* to express smooth variation of the centroids along a topological space. Using a discretized differential

operator matrix  $\mathbf{D}$  on the topological space, the density of this smoothing prior is defined by

$$g(\mathbf{w}|\alpha) = \prod_{j=1}^m \left(\frac{\alpha}{2\pi}\right)^{l/2} (\det^+ \mathbf{D}'\mathbf{D})^{1/2} \exp\left(-\frac{\alpha}{2}\|\mathbf{D}\mathbf{w}_{(j)}\|^2\right) \quad (2.3)$$

where  $\mathbf{w}_{(j)} = (w_{1j}, \dots, w_{rj})'$ ,  $l = \text{rank } \mathbf{D}'\mathbf{D}$  and  $\det^+ \mathbf{D}'\mathbf{D}$  denotes the product of positive eigenvalues of  $\mathbf{D}'\mathbf{D}$ . The hyperparameter  $\alpha$  represents the strength of smooth constraint. Although the elastic net uses the first-order differential operator on a one-dimensional closed-loop topology, we can use various kinds of differential operators and topologies.

From the likelihood (2.1) and the prior (2.3), we obtain the log posterior by Bayes' theorem:

$$\begin{aligned} \log g(\mathbf{w}|\mathbf{X}, \alpha, \beta) &= \log f(\mathbf{X}|\mathbf{w}, \beta) + \log g(\mathbf{w}|\alpha) + \text{const.} \\ &= \sum_{i=1}^n \log \sum_{s=1}^r \exp\left(-\frac{\beta}{2}\|\mathbf{x}_i - \mathbf{w}_s\|^2\right) - \frac{\alpha}{2} \sum_{j=1}^m \|\mathbf{D}\mathbf{w}_{(j)}\|^2 + \text{const.} \end{aligned} \quad (2.4)$$

This corresponds to the negative energy function of an elastic net, whose maximizer gives the MAP estimates of centroids. The elastic net algorithm is a MAP estimation algorithm using the gradient ascent method. We can also use the expectation-maximization (EM) algorithm (Yuille et al., 1994; Utsugi, 1994), which is explained in section 3.

## 2.2 Selection of hyperparameters by empirical Bayesian method

Next, we obtain the marginal likelihood of hyperparameters  $\alpha$  and  $\beta$ :

$$f(\mathbf{X}|\alpha, \beta) = \int f(\mathbf{X}|\mathbf{w}, \beta)g(\mathbf{w}|\alpha)d\mathbf{w} = \int f(\mathbf{w}, \mathbf{X}|\alpha, \beta)d\mathbf{w} \quad (2.5)$$

This is also called the *evidence* of the hyperparameters. Although we desire to obtain the optimal values of hyperparameters by maximizing the evidence, we have difficulty calculating the integral in (2.5) exactly. Here, we use a Gaussian approximation (MacKay, 1992), where the logarithm of integrand is substituted by its quadratic approximation at the maximizer.

In the present case, using the MAP estimate  $\hat{\mathbf{w}}$  and the negative Hesse matrix of the log posterior:

$$\mathbf{H}(\mathbf{w}) = -\frac{\partial^2}{\partial \mathbf{w} \partial \mathbf{w}'} \log g(\mathbf{w}|\mathbf{X}, \alpha, \beta) = -\frac{\partial^2}{\partial \mathbf{w} \partial \mathbf{w}'} \log f(\mathbf{w}, \mathbf{X}|\alpha, \beta) \quad (2.6)$$

the integrand is approximated as

$$f(\mathbf{w}, \mathbf{X}|\alpha, \beta) \simeq f(\hat{\mathbf{w}}, \mathbf{X}|\alpha, \beta) \exp\left(-\frac{1}{2}\mathbf{w}'\mathbf{H}(\hat{\mathbf{w}})\mathbf{w}\right) \quad (2.7)$$

Then, the evidence is approximated as

$$\begin{aligned} f(\mathbf{X}, S_{\hat{\mathbf{w}}}| \alpha, \beta) &= \int_{S_{\hat{\mathbf{w}}}} f(\mathbf{X}|\mathbf{w}, \beta)g(\mathbf{w}|\alpha)d\mathbf{w} \\ &= \int_{S_{\hat{\mathbf{w}}}} f(\mathbf{w}, \mathbf{X}|\alpha, \beta)d\mathbf{w} \\ &\simeq (2\pi)^{rm/2}(\det \mathbf{H}(\hat{\mathbf{w}}))^{-1/2}f(\hat{\mathbf{w}}, \mathbf{X}|\alpha, \beta) \end{aligned} \quad (2.8)$$

where  $S_{\hat{\mathbf{w}}}$  is a region dominated by  $\hat{\mathbf{w}}$  in the parameter space. This evidence consists of probability mass on only  $S_{\hat{\mathbf{w}}}$ , and thus it should be called *local evidence*. We use the local evidence to select the values of hyperparameters, like MacKay's manner for the backpropagation learning.

Now, the log evidence is calculated by

$$\begin{aligned} \log f(\mathbf{X}, S_{\hat{\mathbf{w}}}, |\alpha, \beta) &\simeq \frac{nm}{2} \log \beta + \sum_{i=1}^n \log \sum_{s=1}^r \exp\left(-\frac{\beta}{2}\|\mathbf{x}_i - \hat{\mathbf{w}}_s\|^2\right) + \frac{lm}{2} \log \alpha \\ &\quad - \frac{\alpha}{2} \sum_{j=1}^m \|\mathbf{D}\hat{\mathbf{w}}_{(j)}\|^2 - \frac{1}{2} \log \det \mathbf{H}(\hat{\mathbf{w}}) + \text{const.} \end{aligned} \quad (2.9)$$

The matrix  $\mathbf{H}(\hat{\mathbf{w}})$  is the sum of negative Hesse matrices of the log likelihood and the log prior, which are denoted by  $\mathbf{H}_f(\hat{\mathbf{w}})$  and  $\mathbf{H}_g$ , respectively. The matrix  $\mathbf{H}_f(\hat{\mathbf{w}})$  consists of submatrices:

$$\begin{aligned} \mathbf{H}_{st}(\hat{\mathbf{w}}) &= -\frac{\partial^2}{\partial \mathbf{w}_s \partial \mathbf{w}_t'} \log f(\mathbf{X}|\hat{\mathbf{w}}, \beta) \\ &= \begin{cases} \beta^2 \sum_i^n p_{si}(p_{si} - 1)(\mathbf{x}_i - \hat{\mathbf{w}}_s)(\mathbf{x}_i - \hat{\mathbf{w}}_s)' + \beta n_s \mathbf{I}_m, & s = t \\ \beta^2 \sum_i^n p_{si} p_{ti} (\mathbf{x}_i - \hat{\mathbf{w}}_s)(\mathbf{x}_i - \hat{\mathbf{w}}_t)', & s \neq t \end{cases} \\ &\quad (s, t = 1, \dots, r) \end{aligned} \quad (2.10)$$

where  $\mathbf{I}_m$  is an identity matrix with size  $m$ . The quantity  $p_{si}$  is defined by

$$p_{si} = \frac{f(\mathbf{x}_i|\hat{\mathbf{w}}_s, \beta)}{\sum_{s=1}^r f(\mathbf{x}_i|\hat{\mathbf{w}}_s, \beta)} \quad (2.11)$$

which is called the *fuzzy membership* of the  $i$ th data to the  $s$ th component, and  $n_s = \sum_{i=1}^n p_{si}$  is the estimated number of data points belonging to the  $s$ th component. The matrix  $\mathbf{H}_g$  is given by

$$\mathbf{H}_g = \alpha \mathbf{D}' \mathbf{D} \otimes \mathbf{I}_m \quad (2.12)$$

where “ $\otimes$ ” denotes the Kronecker product.

By maximizing the evidence, we obtain estimates of  $\alpha$  and  $\beta$ . Simulation experiments showed that this method gives good solutions (Utsugi, 1993).<sup>1</sup>

### 3 Derivation of SOM algorithm

In this section, the original SOM algorithm for finite data is derived as an approximate MAP estimation algorithm for the GDM.

Initially, we show that the EM algorithm of GDM is an alternate iteration of smoothing and soft classification. Now, we define binary membership variables  $\mathbf{Y} = (y_{si})$ , where  $y_{si}$  denotes the membership of the  $i$ th data point to the  $s$ th component. Regarding these variables as missing data, we obtain a *complete likelihood*

$$f(\mathbf{X}, \mathbf{Y} | \mathbf{w}, \beta) = \prod_{i=1}^n \prod_{s=1}^r \left( \frac{1}{r} f(\mathbf{x}_i | \mathbf{w}_s, \beta) \right)^{y_{si}} \quad (3.1)$$

which would be an exact likelihood if the missing data were acquired. In the EM algorithm, the following function is maximized instead of the genuine log posterior (2.4):

$$Q(\mathbf{w}) = E_{\mathbf{Y}}(\log f(\mathbf{X}, \mathbf{Y} | \mathbf{w}, \beta) | \mathbf{X}, \hat{\mathbf{w}}, \beta) + \log g(\mathbf{w} | \alpha) \quad (3.2)$$

where  $\hat{\mathbf{w}}$  is a temporary estimate and  $E_{\mathbf{Y}}(\cdot | \cdot)$  denotes conditional expectation with respect to  $\mathbf{Y}$ . The maximizer of  $Q$  is used as  $\hat{\mathbf{w}}$  at the next step. This procedure is iterated until convergence. By calculating  $Q$ , we obtain

$$Q(\mathbf{w}) = -\frac{\beta}{2} \sum_{i=1}^n \sum_{s=1}^r p_{si} \|\mathbf{x}_i - \mathbf{w}_s\|^2 - \frac{\alpha}{2} \sum_{j=1}^m \|\mathbf{D}\mathbf{w}_{(j)}\|^2 + \text{const.} \quad (3.3)$$

where  $p_{si}$  is the fuzzy membership (2.11) using the temporary estimate  $\hat{\mathbf{w}}$ . The maximization of  $Q$  is equivalent to the independent minimizations of

$$H_j(\mathbf{w}_{(j)}) = \sum_{s=1}^r n_s (\bar{x}_{sj} - w_{sj})^2 + \gamma \|\mathbf{D}\mathbf{w}_{(j)}\|^2, \quad (j = 1, \dots, m) \quad (3.4)$$

where  $\bar{x}_{sj}$  is the mean of data weighted by the fuzzy membership:

$$\bar{x}_{sj} = \frac{1}{n_s} \sum_{i=1}^n p_{si} x_{ij} \quad (3.5)$$

---

<sup>1</sup>This evidence is improved from the previously presented one, for which  $r$  was used instead of  $l$  in (2.9). In a wide range of  $\alpha$ , these are almost the same. However, the old evidence grows infinitely as  $\alpha \rightarrow \infty$ , while the new one stays finite.

and  $\gamma = \alpha/\beta$ . Each function  $H_j$  can be regarded as a discretized Laplacian smoothing criterion (O’Sullivan, 1991) for the observations  $\{\bar{x}_{sj} : s = 1, \dots, r\}$  with confidence weights  $\{n_s\}$ , which are obtained through the soft competition process (2.11). A smooth curve minimizing this criterion is used as the next temporary estimate and given by

$$\tilde{w}_{(j)} = \mathbf{K} \mathbf{N} \bar{\mathbf{x}}_{(j)} \quad (3.6)$$

where  $\mathbf{N}$  is a diagonal matrix consisting of  $\{n_s\}$ ,  $\bar{\mathbf{x}}_{(j)} = (\bar{x}_{1j}, \dots, \bar{x}_{rj})'$  and

$$\mathbf{K} = (\mathbf{N} + \gamma \mathbf{D}' \mathbf{D})^{-1} \quad (3.7)$$

From the above, the EM algorithm of GDM can be regarded as an alternate iteration of discretized Laplacian smoothing and soft classification.

On the other hand, Mulier and Cherkassky (1995) interpreted the batch SOM algorithm as an alternate iteration of kernel smoothing and hard classification. Using  $n_t$  and  $\bar{\mathbf{x}}_t = (\bar{x}_{t1}, \dots, \bar{x}_{tm})'$ , which are the number and the mean of data points in the Voronoi region of the weight point of the  $t$ th inner unit, they expressed the weight update rule of SOM as

$$\tilde{w}_{sj} = \sum_{t=1}^r \kappa(s, t) n_t \bar{x}_{tj} / \sum_{t=1}^r \kappa(s, t) n_t, \quad (j = 1, \dots, m) \quad (3.8)$$

where  $\kappa(s, t)$  is a kernel function, for example, a Gaussian density function with respect to  $s - t$  for the normal one-dimensional topology. This is regarded as extended Nadaraya-Watson kernel smoothing (Härdle, 1990) of the observations  $\{\bar{x}_{tj}\}$  with confidence weights  $\{n_t\}$ .

In this point, the difference between GDM and SOM is summarized as follows: (1) Soft classification vs. hard classification. (2) Discretized Laplacian smoothing vs. kernel smoothing. (3) In the original SOM, incremental learning is used rather than batch learning. Each method used in SOM can be regarded as an approximation of the associated method used in GDM, as explained below.

The soft competition process (2.11) turns hard as  $\beta \rightarrow \infty$ , and thus hard classification gives good approximation for soft classification if  $\beta$  is large.<sup>2</sup>

The discretized Laplacian smoothing can also be approximated by kernel smoothing. As shown in the appendix, a curve smoothed by discretized Laplacian smoothing (3.6) is expressed by the kernel smoothing form (3.8) using the entries of  $\mathbf{K}$  as the values of kernel function  $\kappa(s, t)$ . In reality, this kernel function is variable

---

<sup>2</sup>Also, hard competition can be derived from a MAP estimation algorithm of GDM with a classification likelihood rather than the mixture likelihood (2.1). The *classification likelihood* has the same form as the complete likelihood (3.1), though  $\mathbf{Y}$  is regarded as a parameter rather than missing data. In general, classification likelihood approaches lead to poorer results than mixture likelihood approaches (McLachlan and Basford, 1988).

according to the variation of  $\mathbf{N}$ , unlike SOM. In many cases, however,  $\mathbf{N}$  is nearly proportional to  $\mathbf{I}_r$  at the last stage of learning, since the expectation of  $\mathbf{N}$  is proportional to  $\mathbf{I}_r$ . In particular, for the second-order differential operator on a one-dimensional line-segment topology, whose entries are

$$d_{ij} = \begin{cases} -2 & |i - j + 1| = 0 \\ 1 & |i - j + 1| = 1 \\ 0 & \textit{otherwise} \end{cases} \quad (i = 1, \dots, r - 2; j = 1, \dots, r). \quad (3.9)$$

we can obtain an explicit form of the kernel function using Silverman's (1984) equivalent kernel of spline smoothing (Utsugi, 1994). This kernel function has a Mexican hat shape with a width proportional to  $\gamma^{1/4}$ .

Finally, we can also use a generalized EM algorithm (Dempster et al., 1977), where we use

$$\mathbf{w} = (1 - c)\hat{\mathbf{w}} + c\tilde{\mathbf{w}} \quad (0 < c \leq 1) \quad (3.10)$$

as the next temporary estimate instead of  $\tilde{\mathbf{w}}$ . Using small  $c$ , we can make the variation of the parameter in one leaning step arbitrary small, where batch and incremental leaning is similar.

## 4 Comparison of hyperparameter selection methods

### 4.1 Hyperparameter selection by cross-validation

In section 2, an empirical Bayesian method for hyperparameter selection was studied. Another commonly used method for such problems is cross-validation. In the present section, we apply cross-validation to our problem and compare the results with those of the empirical Bayesian method through simulation experiments.

Generally, the rationale of cross-validation is as follows. The expected log likelihood (ELL) by a true data distribution is a good measure of model adequacy. In reality, we cannot calculate this value because of the unknown true data distribution. Instead, we use a cross-validation score as an unbiased estimate of ELL. However, this estimated criterion has considerable variance for small data, and its maximizer is not an unbiased estimate for the peak position of ELL. In the next simulation, we will calculate ELL, in addition to cross-validation scores and evidence, to observe the bias and variance of these criteria.

### 4.2 Simulation experiments

We apply a GDM with 20 components and the differential operator (3.9) to two types of artificial data sets: low and high noise condition (Figure 1). By using

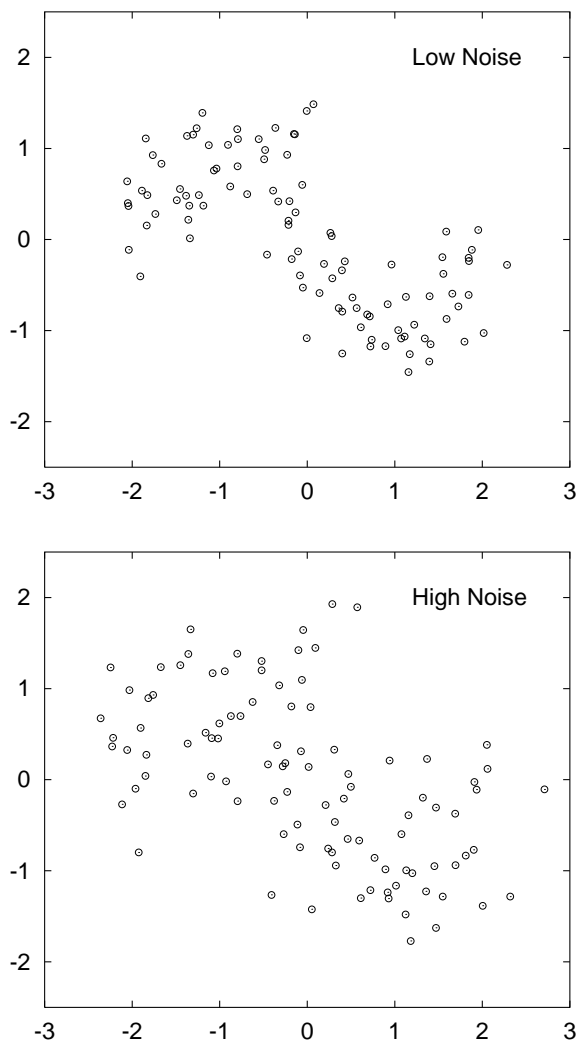


Figure 1: Scatter plots for samples of data sets.

Artificial data  $\mathbf{x}_i = (x_{i1}, x_{i2})'$ , ( $i = 1, \dots, 100$ ) are generated from two independent standard Gaussian random series  $\{e_{i1}\}$  and  $\{e_{i2}\}$  by  
 $x_{i1} = 4(i-1)/n - 2 + \sigma e_{i1}$ ,  $x_{i2} = \sin[2\pi(i-1)/n] + \sigma e_{i2}$ . Two conditions of noise level are used: low noise ( $\sigma = 0.3$ ) and high noise ( $\sigma = 0.5$ ). For each condition, 20 data sets are used in the simulation.

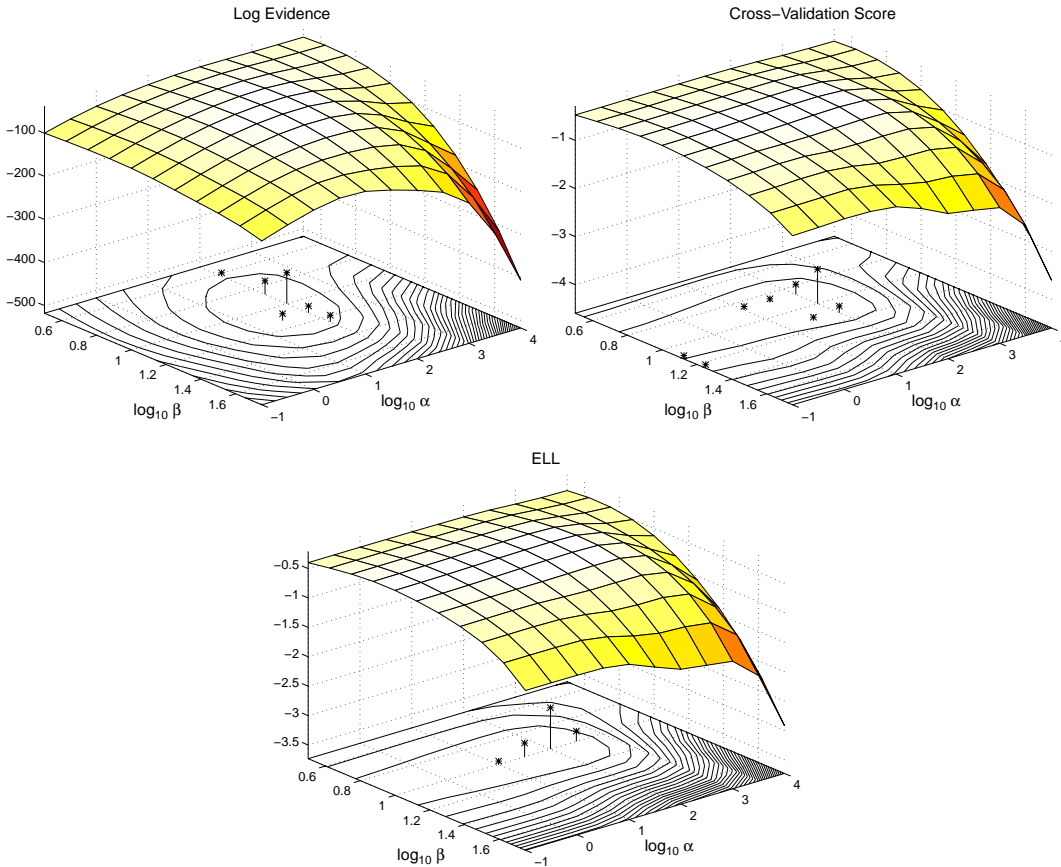


Figure 2: Landscapes and contour maps of averaged criteria and peak-position histograms for 20 data sets in the low noise condition.

For each data set, the MAP estimates of centroids are obtained by the EM algorithm of GDM ( $r = 20$ ). The values of hyperparameters are taken from the grid points in the hyperparameter space. For each fixed  $\beta$ ,  $\alpha$  is decremented in the grid, and the MAP estimates in the preceding  $\alpha$  are used as initial values of centroids. Log evidence is calculated by (2.9). Cross-validation scores are obtained in the following manner. A data set is divided randomly into 10 groups of the same size. For each group, centroids are re-estimated using the data in all but the group, where the MAP estimates from all data are used as initial values. For the re-estimated centroids, their log likelihood is calculated using the data in the group. A cross-validation score is given as the mean of the log likelihoods. ELL is approximated by the mean log likelihood of the MAP estimates evaluated by 1000 newly generated data.

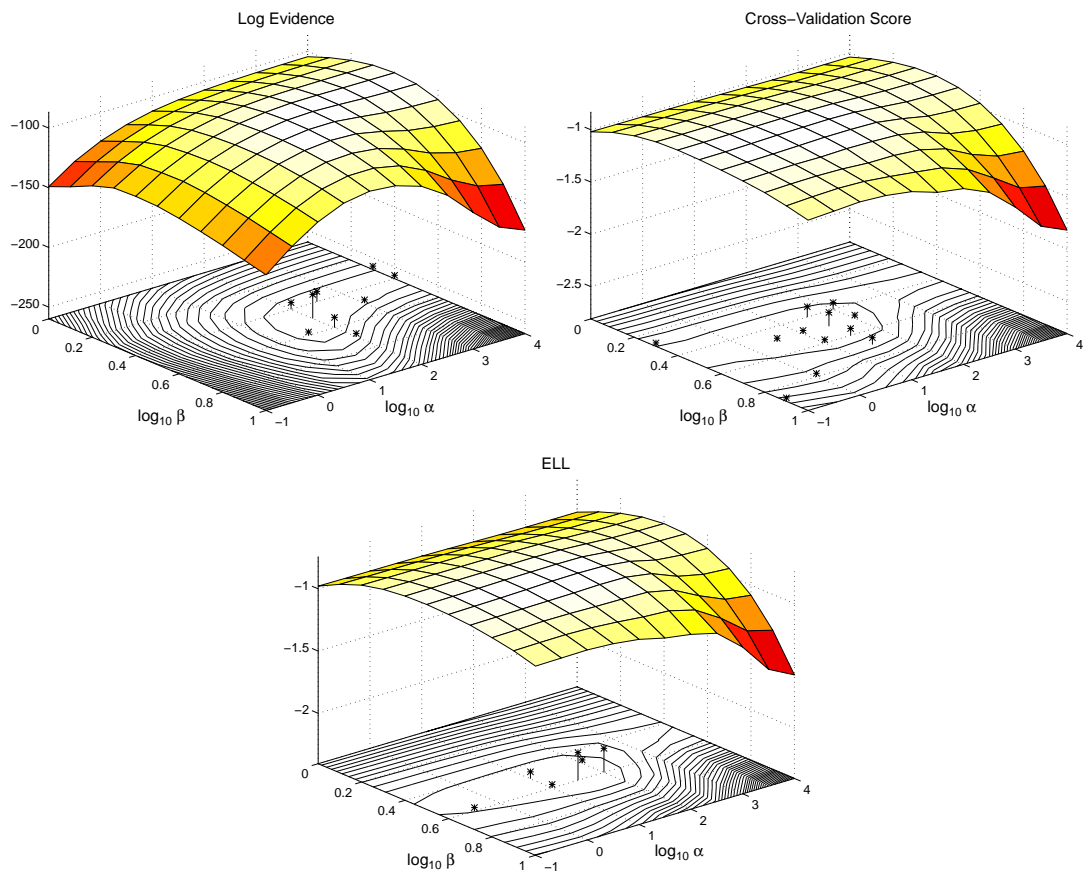


Figure 3: Landscapes and contour maps for averaged criteria and peak position histograms in the high noise condition.

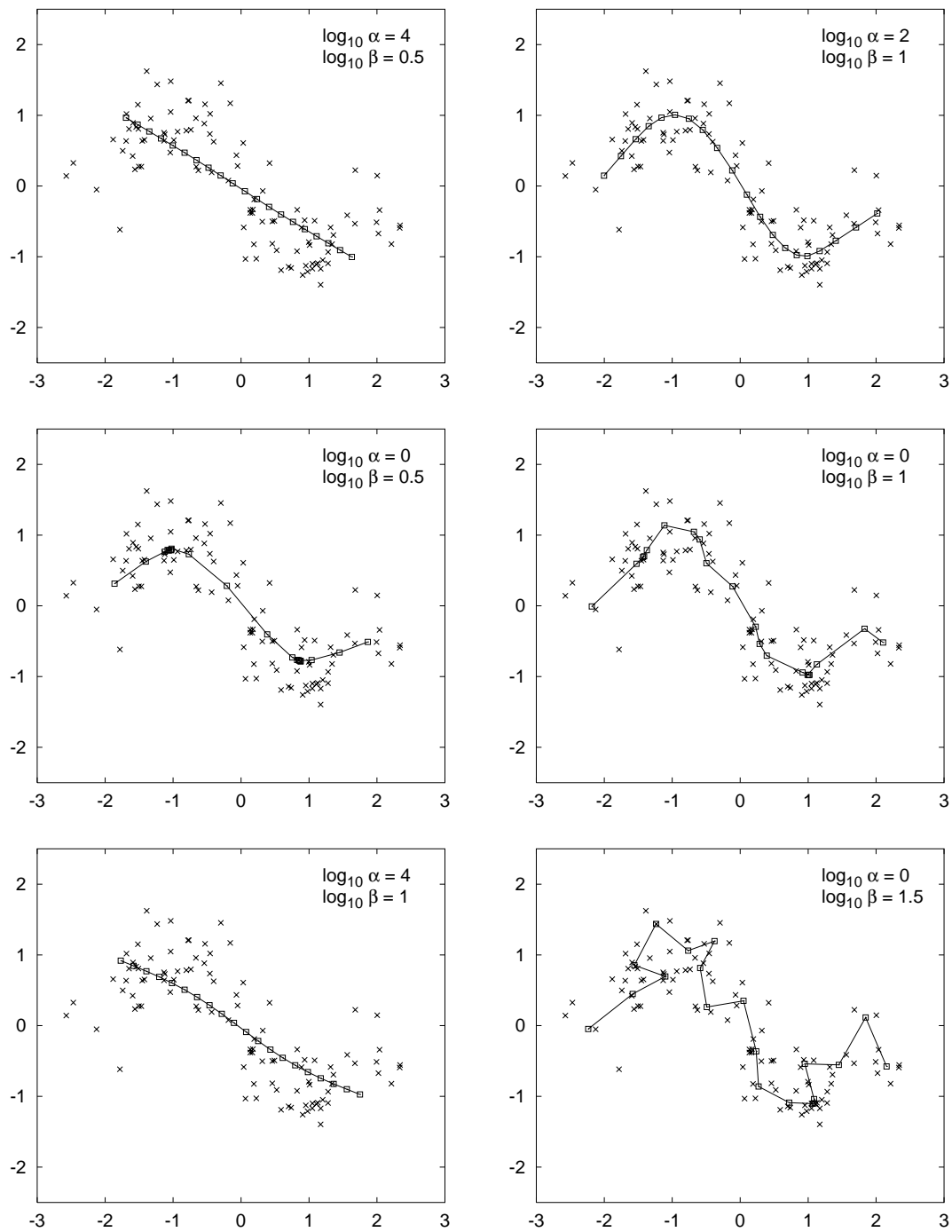


Figure 4: Samples of centroid configurations for several hyperparameter values in the low noise condition.

the values of hyperparameters at rectangular grid points in the hyperparameter space, we obtain the graph of each criterion. The landscapes and contour maps in Figures 2 and 3 are obtained by averaging each criterion for 20 different data sets. The histograms of peak positions for individual data sets also are shown in the figures. Figure 4 illustrates the configurations of estimated centroid parameters under several values of  $(\alpha, \beta)$  in the low noise condition. From these figures, we can conclude the following.

In the case of low noise (Figure 2), peak positions of all criteria are close to each other except for few peaks of cross-validation scores. In the area where the majority of peaks are gathering, the configurations of centroid parameters have good appearances (Figure 4,  $\log_{10} \alpha = 2, \log_{10} \beta = 1$ ). Thus, we can say that the both methods succeed for the most part. However, cross-validation scores have two peaks with much lower  $\alpha$  than the other peaks, which lead to too complicated configurations. This is probably due to the flatness of the averaged landscape in its low  $\alpha$  area and large variance of cross-validation scores. Because of this instability, the cross-validation method is inferior to the other method in this case, though its averaged landscape is similar to that of ELL.

In the case of high noise (Figure 3), the discrepancies among the criteria increase. While cross-validation has a bias toward low  $\alpha$  again, evidence comes to have the opposite bias. In this case, we have difficulty choosing between the methods. However, the property that evidence leads to the simplest model unless sufficient data for structure determination are given may be desirable because it agrees with a general strategy of data analysis that linear models should be used for very noisy data rather than nonlinear ones.

## 5 Conclusion

We derived the SOM algorithm as an approximate MAP estimation algorithm for a GDM. Then, several methods to evaluate the quality of hyperparameters for this model were developed by empirical Bayesian estimation and cross-validation. These methods were compared through simulation studies. It was found that the cross-validation methods favor complex structures, while the empirical Bayesian methods favor simple ones. Because of these properties and the long calculation time for cross-validation, the empirical Bayesian methods are recommended for this model.

## References

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from

- incomplete data via the EM algorithm. *J. Roy. Statist. Soc., Ser. B*, 39:1–38.
- Durbin, R., Szeliski, R., and Yuille, A. (1989). An analysis of the elastic net approach to the traveling salesman problem. *Neural Comp.*, 1:348–358.
- Härdle, W. (1990). *Smoothing Techniques with Implementation in S*. Springer-Verlag, Berlin.
- Kohonen, T. (1988). *Self-Organization and Associative Memory (2nd ed.)*. Springer-Verlag, Berlin.
- Kohonen, T. (1990). The self-organizing map. *Proc. IEEE*, 78:1464–1480.
- MacKay, D. J. C. (1992). A practical Bayesian framework for backprop networks. *Neural Comp.*, 4:448–472.
- McLachlan, G. J. and Basford, K. E. (1988). *Mixture models: Inference and Applications to Clustering*. Marcel Dekker, New York.
- Mulier, F. and Cherkassky, V. (1995). Self-organization as an iterative kernel smoothing process. *Neural Comp.*, 7:1141–1153.
- Nowlan, S. J. (1990). Maximum likelihood competitive learning. In *Advances in Neural Information Processing Systems 2*, pages 574–582. Morgan Kaufmann, San Mateo, CA.
- O’Sullivan, F. (1991). Discretized Laplacian smoothing by Fourier methods. *J. Amer. Statist. Assoc.*, 86:634–642.
- Silverman, B. W. (1984). Spline smoothing: The equivalent variable kernel method. *Ann. Statist.*, 12:898–916.
- Utsugi, A. (1993). A Bayesian model of topology-preserving map learning (in Japanese). *Trans. IEICE D-II*, J76-D-II:1232–1239.
- Utsugi, A. (1994). Lateral interaction in Bayesian self-organizing maps (in Japanese). *Trans. IEICE D-II*, J77-D-II:1329–1336.
- Yuille, A. L. (1990). Generalized deformable models, statistical physics, and matching problems. *Neural Comp.*, 2:1–24.
- Yuille, A. L., Stolorz, P., and Utans, J. (1994). Statistical physics, mixture of distributions and the EM algorithm. *Neural Comp.*, 6:334–340.

## Appendix

In this appendix, we consider the relation between kernel smoothing and discretized Laplacian smoothing.

Initially, we show

$$\mathbf{KN}\mathbf{1}_r = \mathbf{1}_r \tag{A.1}$$

where  $\mathbf{1}_r$  is a  $r$ -dimensional column vector with all ones. In general,  $\mathbf{D}\mathbf{1}_r = \mathbf{0}$  since  $\mathbf{D}$  is a differential operator. Thus, using (3.7)

$$(\mathbf{KN})^{-1}\mathbf{1}_r = (\mathbf{I}_r + \gamma\mathbf{N}^{-1}\mathbf{D}'\mathbf{D})\mathbf{1}_r = \mathbf{1}_r \tag{A.2}$$

This means that  $(\mathbf{KN})^{-1}$  has  $\mathbf{1}_r$  as an eigenvector with the eigenvalue one, thus  $\mathbf{KN}$  also have the same property. This leads to (A.1).

From (3.6) and (A.1), a curve smoothed by discretized Laplacian smoothing is expressed by the kernel smoothing form (3.8) using the entries of  $\mathbf{K}$  as the values of kernel function  $\kappa(s, t)$ .