



University
of Glasgow

Husmeier, D. (2000) The Bayesian evidence scheme for regularizing probability-density estimating neural networks. *Neural Computation*, 12 (11). pp. 2685-2717. ISSN 0899-7667

Copyright © 2000 Massachusetts Institute of Technology.

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

The content must not be changed in any way or reproduced in any format or medium without the formal permission of the copyright holder(s)

When referring to this work, full bibliographic details must be given

<http://eprints.gla.ac.uk/85654/>

Deposited on: 11 September 2013

Enlighten – Research publications by members of the University of Glasgow
<http://eprints.gla.ac.uk>

The Bayesian Evidence Scheme for Regularizing Probability-Density Estimating Neural Networks

Dirk Husmeier

Biomathematics and Statistics Scotland, Scottish Crop Research Institute, Invergowrie, Dundee DD2 5DA, U.K.

Training probability-density estimating neural networks with the expectation-maximization (EM) algorithm aims to maximize the likelihood of the training set and therefore leads to overfitting for sparse data. In this article, a regularization method for mixture models with generalized linear kernel centers is proposed, which adopts the Bayesian evidence approach and optimizes the hyperparameters of the prior by type II maximum likelihood. This includes a marginalization over the parameters, which is done by Laplace approximation and requires the derivation of the Hessian of the log-likelihood function. The incorporation of this approach into the standard training scheme leads to a modified form of the EM algorithm, which includes a regularization term and adapts the hyperparameters on-line after each EM cycle. The article presents applications of this scheme to classification problems, the prediction of stochastic time series, and latent space models.

1 Introduction and Notation ---

Consider the problem of inferring the probability density $P(\mathbf{y}_t|\mathbf{x}_t)$ of some m -dimensional target vector \mathbf{y}_t conditional on an n -dimensional vector of explanatory variables \mathbf{x}_t . A common approach is to approximate the unknown true distribution by a mixture model (Jacobs, Jordan, Nowlan, & Hinton, 1991; Jordan & Jacobs, 1994; Bishop, 1995), for which, in this article, the functional form of Husmeier and Taylor (1998) will be chosen. Let $\mathbf{g}(\mathbf{x}_t) \in \mathbb{R}^{\tilde{n}}$ denote a (fixed and in general nonlinear) transformation of the explanatory variables, and define the following generalized linear function,

$$f_{ki}(\mathbf{x}_t) := f(\mathbf{x}_t; \mathbf{w}_{ki}) := \mathbf{w}_{ki}^\dagger \mathbf{g}(\mathbf{x}_t) \quad (1.1)$$

in which \mathbf{w}_{ki} is an \tilde{n} -dimensional parameter vector. (Note that the dimensions of $\mathbf{g}(\mathbf{x}_t)$ and \mathbf{x}_t are in general different: $\tilde{n} \neq n$.) Also, introduce the positive and normalized mixing coefficients p_k ($p_k \geq 0$, $\sum_k p_k = 1$) and the positive precisions (inverse variances) $\beta_k > 0$. Then a multivariate generalization of the model discussed in Husmeier and Taylor (1998) has the

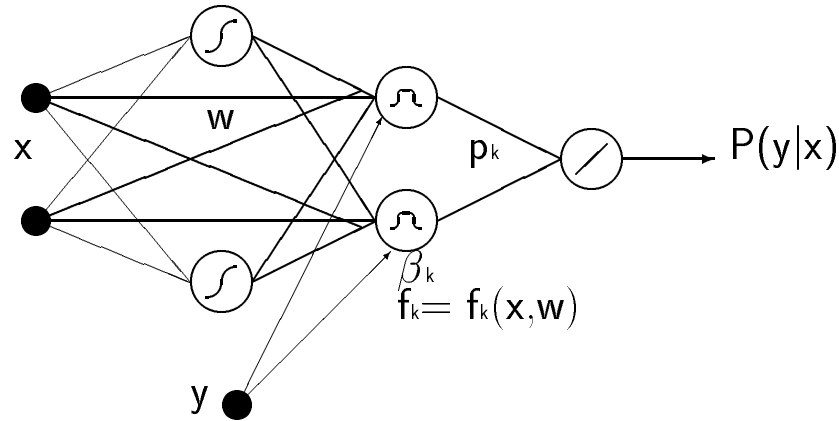


Figure 1: Gaussian mixture network for predicting conditional probability densities. The network contains two hidden layers, where units in the first hidden layer have a sigmoidal and those in the second hidden layer a gaussian transfer function. The precisions of the gaussian bumps, β_k , are treated as adaptable parameters, and the output weights p_k are positive and normalized. Arrows symbolize weights with a constant value of one; bold lines represent wires with adaptable weights. When applying the random vector functional link net approach (discussed in the text), the weights between the input and the first hidden layer, plotted as narrow lines, are drawn at random from some appropriate distribution and then remain fixed during training.

following form:

$$P(\mathbf{y}_t|\mathbf{x}_t) = \sum_{k=1}^K p_k P(\mathbf{y}_t|\mathbf{x}_t, k) \quad (1.2)$$

$$P(\mathbf{y}_t|\mathbf{x}_t, k) = \sqrt{\prod_{i=1}^m \frac{\beta_{ki}}{2\pi}} \exp\left(-\sum_{i=1}^m \frac{\beta_{ki}}{2} [y_{ti} - f_{ki}(\mathbf{x}_t)]^2\right). \quad (1.3)$$

A possible neural network realization is shown in Figure 1. Note that the restriction implied by equation 1.1 is essentially a generalized linear model. This allows the model parameters

$$\mathbf{q} := \{p_1, \dots, p_k, \dots, p_{K-1}, \beta_{11}, \dots, \beta_{ki}, \dots, \beta_{Km}, \mathbf{w}_{11}, \dots, \mathbf{w}_{ki}, \dots, \mathbf{w}_{Km}\} \quad (1.4)$$

to be adapted with the expectation-maximization (EM) algorithm (Demp-

ster, Laird, & Rubin, 1977) which is known to be a fast (superlinear) algorithm to maximize the likelihood $P(\mathbf{D}|\mathbf{q})$ of the training set $\mathbf{D} := \{\mathbf{x}_t, \mathbf{y}_t\}_{t=1}^N$ (see, e.g., Xu & Jordan, 1996). Here and in what follows, lowercase boldface letters denote column vectors, uppercase boldface letters denote matrices, and the dagger (\dagger) indicates matrix transposition. The index $k \in \{1, \dots, K\}$ will be used to label kernels, $t \in \{1, \dots, N\}$ labels training exemplars, and the subscript $i \in \{1, \dots, m\}$ in y_{ti} represents the i th coordinate of the target vector \mathbf{y}_t . The mixing coefficients and kernel precisions are collected in the vectors $\mathbf{p} := (p_1, \dots, p_{K-1})^\dagger$ and $\beta := (\beta_{11}, \dots, \beta_{Km})^\dagger$.¹ The vector $\mathbf{w}_k := (\mathbf{w}_{k1}^\dagger, \dots, \mathbf{w}_{km}^\dagger)^\dagger$ comprises all the weights feeding into the k th kernel. Alternatively, the matrix notation $\mathbf{W}_k := (\mathbf{w}_{k1}, \dots, \mathbf{w}_{km})^\dagger$ will occasionally be used (see equation 2.18).² Finally, \mathbf{w} denotes the vector of *all* weights in the network: $\mathbf{w} := (\mathbf{w}_1^\dagger, \dots, \mathbf{w}_K^\dagger)^\dagger$.

2 Regularization with the Bayesian Evidence Scheme

2.1 Summary of the Concept. It is well known that for sparse training data, a maximum likelihood approach usually leads to severe overfitting. The objective of this article, therefore is to generalize the Bayesian evidence scheme (MacKay, 1992) to regularizing probability-estimating neural networks. First, introduce a gaussian prior on the weights \mathbf{w} , which depends on a vector of hyperparameters $\alpha = (\alpha_1, \dots, \alpha_K)^\dagger$:

$$P(\mathbf{w}|\alpha) := \prod_{k=1}^K \prod_{i=1}^m P(\mathbf{w}_{ki}|\alpha_k) := \prod_{k=1}^K \prod_{i=1}^m \left(\frac{\alpha_k}{2\pi}\right)^{\tilde{n}/2} \exp\left(-\frac{\alpha_k}{2} \mathbf{w}_{ki}^\dagger \mathbf{w}_{ki}\right). \quad (2.1)$$

This is equivalent to the common regularization method of linear weight decay, by which large values of \mathbf{w}_{ki} are penalized. Note that equation 2.1 implies a division of the weights \mathbf{w} into several weight groups \mathbf{w}_{ki} , with weights feeding into the same kernel sharing a common weight-decay hyperparameter α_k . Also, note that the hierarchical structure of the model, $\alpha \rightarrow \mathbf{w} \rightarrow \mathbf{y}$, implies conditional independence of \mathbf{y} on α given \mathbf{w} . On observing the training data $\mathbf{D} = \{\mathbf{y}_t, \mathbf{x}_t\}_{t=1}^N$, the most probable parameter values are those that maximize the posterior probability,

$$P(\mathbf{w}, \beta, \mathbf{p}, \alpha|\mathbf{D}) \propto P(\mathbf{D}|\mathbf{w}, \beta, \mathbf{p})P(\mathbf{w}|\alpha), \quad (2.2)$$

where the indicated proportionality holds for a metric in which the prior

¹ Note that due to the constraint $\sum_k p_k = 1$, only $K-1$ mixing coefficients are adaptable parameters.

² Note that \mathbf{w}_k is an $m\tilde{n}$ -dimensional vector, while \mathbf{W}_k is an m -by- \tilde{n} matrix.

$P(\beta, \mathbf{p}, \alpha)$ is uniform. Define

$$E_o(\mathbf{w}, \beta, \mathbf{p}) := -\ln P(\mathbf{D}|\mathbf{w}, \beta, \mathbf{p}) \quad (2.3)$$

$$R(\mathbf{w}; \alpha) := -\ln P(\mathbf{w}|\alpha) \quad (2.4)$$

$$E(\mathbf{w}, \beta, \mathbf{p}; \alpha) := E_o(\mathbf{w}, \beta, \mathbf{p}) + R(\mathbf{w}; \alpha). \quad (2.5)$$

Note that unregularized training corresponds to minimizing the cost function E_o , whereas regularized training with fixed weight-decay hyperparameters α is effected by minimizing the total cost function E . Since, in general, we do not have sufficient prior knowledge to decide on the values for α in advance, a straightforward approach seems to be the maximization of the posterior probability in equation 2.2 with respect to all parameters. However, as pointed out by MacKay (1992), this is likely to lead to suboptimal results: the joint posterior distribution tends to be strongly skewed with a mode that is not representative of the distribution as a whole (see also Bishop & Qazaz, 1995, and MacKay, 1993, for further discussion). This article therefore follows the idea in MacKay (1992) to find the mode of $P(\beta, \mathbf{p}, \alpha|\mathbf{D})$ after integrating out the weight parameters \mathbf{w} . Assuming again a metric in which $P(\beta, \mathbf{p}, \alpha)$ is uniform, this is equivalent to maximizing the likelihood

$$\begin{aligned} P(\mathbf{D}|\beta, \mathbf{p}, \alpha) &= \int P(\mathbf{D}, \mathbf{w}|\beta, \mathbf{p}, \alpha) d\mathbf{w} = \int P(\mathbf{D}|\mathbf{w}, \beta, \mathbf{p}) P(\mathbf{w}|\alpha) d\mathbf{w} \\ &= \int \exp[-E(\mathbf{w}; \beta, \mathbf{p}, \alpha)] d\mathbf{w}, \end{aligned} \quad (2.6)$$

where in the last step, equations 2.3 through 2.5 have been applied. Note that MacKay's approach is equivalent to the type II maximum likelihood method of conventional statistics. The integral in equation 2.6 is solved by Laplace approximation, which is equivalent to a Taylor series expansion of E up to second order:

$$E(\mathbf{w}; \beta, \mathbf{p}, \alpha) = E(\hat{\mathbf{w}}; \beta, \mathbf{p}, \alpha) + \frac{1}{2}(\mathbf{w} - \hat{\mathbf{w}})^\dagger \mathbf{H}(\mathbf{w} - \hat{\mathbf{w}}) \quad (2.7)$$

$$\hat{\mathbf{w}} := \operatorname{argmin}_{\mathbf{w}} \{E(\mathbf{w}; \beta, \mathbf{p}, \alpha)\} \quad (2.8)$$

$$\mathbf{H} := \left[\nabla_{\mathbf{w}} \nabla_{\mathbf{w}}^\dagger E \right]_{\mathbf{w}=\hat{\mathbf{w}}}. \quad (2.9)$$

Inserting equation 2.7 into 2.6 and defining $M := \dim \mathbf{w} = m\tilde{n}$ gives:

$$P(\mathbf{D}|\beta, \mathbf{p}, \alpha) = \exp[-E(\hat{\mathbf{w}}; \beta, \mathbf{p}, \alpha)] \sqrt{\frac{(2\pi)^M}{\det \mathbf{H}}} \quad (2.10)$$

$$\begin{aligned} E^*(\hat{\mathbf{w}}; \beta, \mathbf{p}, \alpha) &:= -\ln P(\mathbf{D}|\beta, \mathbf{p}, \alpha) \\ &= E(\hat{\mathbf{w}}; \beta, \mathbf{p}, \alpha) + \frac{1}{2} \ln \det \mathbf{H} - \frac{M}{2} \ln(2\pi). \end{aligned} \quad (2.11)$$

This leads to the following iterative optimization scheme:

1. Given preliminary values for the hyperparameters³ $\beta, \mathbf{p}, \alpha$, minimize the cost function E in equation 2.5 with respect to \mathbf{w} . From equation 2.5, this is equivalent to

$$-\nabla_{\mathbf{w}_{ki}} E_o = \nabla_{\mathbf{w}_{ki}} R. \tag{2.12}$$

2. Given the new values of the weight parameters, $\mathbf{w} = \hat{\mathbf{w}}$, minimize the cost function E^* in equation 2.11 with respect to $\beta, \mathbf{p}, \alpha$. Making use of equations 2.5 and 2.11, this gives

$$\frac{\partial E_o}{\partial r} + \frac{\partial R}{\partial r} = -\frac{1}{2} \frac{\partial}{\partial r} \ln \det \mathbf{H}, \tag{2.13}$$

where r represents any of the hyperparameters p_k, β_{ki} , and α_{ki} .

This scheme is iterated until a self-consistent solution for $\mathbf{w}, \beta, \mathbf{p}, \alpha$ has been found.

2.2 The Regularized EM Algorithm. Updating the parameters and hyperparameters according to equations 2.12 and 2.13 can be accomplished with a modified version of the EM algorithm. This section gives an algorithmic description of this scheme; the detailed derivation is relegated to the appendix. First, define the posterior probability for the generation of data point $(\mathbf{x}_t, \mathbf{y}_t)$ by the k th kernel of the mixture model, equation 1.2:

$$\pi_k(t) := P(k|\mathbf{y}_t, \mathbf{x}_t) = \frac{P(\mathbf{y}_t|\mathbf{x}_t, k)p_k}{\sum_k P(\mathbf{y}_t|\mathbf{x}_t, k)p_k}, \tag{2.14}$$

where Bayes' rule has been applied. Moreover, the following definitions are introduced:

$$\mathbf{G} := (\mathbf{g}(\mathbf{x}_1), \dots, \mathbf{g}(\mathbf{x}_N)) \quad \tilde{n}\text{-by-}N \text{ matrix} \tag{2.15}$$

$$\mathbf{\Pi}_k : \text{diagonal } N\text{-by-}N \text{ matrix with } (\mathbf{\Pi}_k)_{tt'} := \pi_k(t)\delta_{tt'} \tag{2.16}$$

$$\mathbf{I} : \text{unit matrix, } \mathbf{I}_{ij} := \delta_{ij} \tag{2.17}$$

$$\mathbf{W}_k := (\mathbf{w}_{k1}, \dots, \mathbf{w}_{km})^\dagger \tag{2.18}$$

$$\mathbf{y}_t := \begin{pmatrix} y_{t,i=1} \\ \vdots \\ y_{t,i=m} \end{pmatrix}, \quad \mathbf{y}_{.i} := \begin{pmatrix} y_{t=1,i} \\ \vdots \\ y_{t=N,i} \end{pmatrix},$$

³ Strictly speaking, β and \mathbf{p} are parameters treated like hyperparameters.

$$\mathbf{Y} := (\mathbf{y}_{t=1}, \dots, \mathbf{y}_{t=N}) = \begin{pmatrix} \mathbf{y}_{i=1}^\dagger \\ \vdots \\ \mathbf{y}_{i=m}^\dagger \end{pmatrix}. \quad (2.19)$$

Let ε_{ki}^v denote the eigenvalues of the matrix $\mathbf{A}_{ki} := [\nabla_{\mathbf{w}_{ki}} \nabla_{\mathbf{w}_{ki}}^\dagger E_\theta]_{\mathbf{w}=\hat{\mathbf{w}}}$ (for which an explicit expression will be derived later in equation A.28), and introduce, similarly to MacKay (1992), the number of well-determined parameters:

$$\gamma_{ki} := \sum_{v=1}^{\tilde{n}} \frac{\varepsilon_{ki}^v}{\alpha_k + \varepsilon_{ki}^v}, \quad \gamma_k := \sum_{i=1}^m \gamma_{ki}. \quad (2.20)$$

Now, the new regularized EM algorithm is given by the following iterative update equations:

$$\hat{p}_k = \frac{N_k}{N} \quad (2.21)$$

$$\mathbf{G}\Pi_k \mathbf{y}_{.i} = \left(\mathbf{G}\Pi_k \mathbf{G}^\dagger + \frac{\alpha_k}{\beta_{ki}} \mathbf{I} \right) \hat{\mathbf{w}}_{ki} \quad (2.22)$$

$$\frac{1}{\hat{\beta}_{ki}} = \frac{\sum_t \pi_k(t) [f(\mathbf{x}_t, \hat{\mathbf{w}}_k) - y_{ti}]^2}{N_k - \gamma_{ki}} \quad (2.23)$$

$$\frac{1}{\hat{\alpha}_k} = \frac{1}{\gamma_k} \text{tr} \left[\hat{\mathbf{W}}_k \hat{\mathbf{W}}_k^\dagger \right] \quad (2.24)$$

where

$$N_k := \sum_{t=1}^N \pi_k(t) \quad (2.25)$$

and the posterior probabilities $\pi_k(t)$ (defined in equation 2.14) are computed on the basis of the old parameter values. This scheme is iterated according to the standard EM paradigm, where at each iteration the Hessian and its eigenvalues are recalculated, thereby obtaining new values for the numbers of well-determined parameters γ_{ki} according to equation 2.20. Note that the standard unregularized EM algorithm is regained by setting $\alpha_k \equiv 0$ and $\gamma_{ki} \equiv 0$, as seen from a comparison with the respective update equations in Husmeier and Taylor (1998).

Since $N_k = \sum_{t=1}^N \pi_k(t)$ occurs in the denominator of the update equation, 2.23, the algorithm becomes unstable for small N_k , that is, for small mixing coefficients $p_k = N_k/N$. This typically happens when the number of training exemplars N is too small for the given network complexity. It is therefore reasonable to introduce an explicit pruning scheme and discard kernels for which $N_k \leq \gamma_{ki} + \varepsilon$, where in this study $\varepsilon := 10^{-6}$ was chosen.

3 Application to Latent Space Models

Recently latent space models have become very popular, where dependencies between observations in a high-dimensional data space are explained by a smaller number of so-called latent degrees of freedom. This allows the probabilistic reformulation and reinterpretation of several well-established machine learning algorithms. Examples are probabilistic principal component analysis (Tipping & Bishop, 1999), independent factor analysis (Attias, 1999), and the generative topographic map (Bishop, Svensen, & Williams, 1998). This study will show how the Bayesian evidence scheme can be applied to mixtures of probabilistic principal component analyzers (MPPCA), as introduced in Tipping and Bishop (1999).

Two simplifications are inherent in the MPPCA approach. First, the function \mathbf{g} in equation 1.1 is linear, that is,

$$\mathbf{f}_k(\mathbf{x}_t) := \mathbf{W}_k \mathbf{x}_t + \boldsymbol{\mu}_k, \quad (3.1)$$

where $\boldsymbol{\mu}_k$ is a vector of bias parameters.⁴ Second, the gaussian kernels $P(\mathbf{y}_t | \mathbf{x}_t, k)$ are isotropic, that is, $\beta_{ki} = \beta_k \forall i$. Consequently, equation 1.3 simplifies to

$$P(\mathbf{y}_t | \mathbf{x}_t, k) = \left(\frac{\beta_k}{2\pi} \right)^{m/2} \exp \left(-\frac{\beta_k}{2} \|\mathbf{y}_t - \mathbf{W}_k \mathbf{x}_t - \boldsymbol{\mu}_k\|^2 \right), \quad (3.2)$$

and the regularized EM update equations, 2.22 and 2.23, can be written in the more compact form (recall the definitions 2.15–2.20, and see the appendix for a derivation):

$$\mathbf{G} \boldsymbol{\Pi}_k \mathbf{Y}^\dagger = \left(\mathbf{G} \boldsymbol{\Pi}_k \mathbf{G}^\dagger + \frac{\alpha_k}{\beta_k} \mathbf{I} \right) \hat{\mathbf{W}}_k^\dagger \quad (3.3)$$

$$\frac{1}{\hat{\beta}_k} = \frac{\sum_{t=1}^N \pi_k(t) [\mathbf{y}_t - \mathbf{f}_k(\mathbf{x}_t)]^2}{mN_k - \gamma_k}. \quad (3.4)$$

The crucial assumption of the MPPCA model is a normal prior on the latent variables, defined by

$$P(\mathbf{x}_t) = \left(\frac{1}{2\pi} \right)^{n/2} \exp \left(-\frac{1}{2} \mathbf{x}_t^\dagger \mathbf{x}_t \right). \quad (3.5)$$

⁴ Note that this need not be made explicit in equation 1.1 since a bias is easily included by defining one component of $\mathbf{g}(\mathbf{x}_t)$ to be the constant function, $g_i(\mathbf{x}_t) \equiv 1$.

This allows the integration over the latent space to be carried out analytically and leads to a marginal distribution of \mathbf{y}_t in the form

$$P(\mathbf{y}_t) = \sum_{k=1}^K p_k P(\mathbf{y}_t | k) \quad (3.6)$$

$$\begin{aligned} P(\mathbf{y}_t | k) &= \int P(\mathbf{y}_t | \mathbf{x}_t, k) P(\mathbf{x}_t) d\mathbf{x}_t \\ &= [(2\pi)^m \det \mathbf{C}_k]^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{y}_t - \boldsymbol{\mu}_k)^\dagger \mathbf{C}_k^{-1} (\mathbf{y}_t - \boldsymbol{\mu}_k)\right), \end{aligned} \quad (3.7)$$

where \mathbf{C}_k is the so-called model covariance matrix, defined by

$$\mathbf{C}_k := \beta_k^{-1} \mathbf{I} + \mathbf{W}_k \mathbf{W}_k^\dagger. \quad (3.8)$$

The probability of the latent variables \mathbf{x}_t conditional on the data vectors \mathbf{y}_t (needed for data compression) is given by Tipping and Bishop (1999),

$$\begin{aligned} P(\mathbf{x}_t | \mathbf{y}_t, k) &= \sqrt{\left(\frac{\beta_k}{2\pi}\right)^n \det \mathbf{M}_k} \\ &\quad \times \exp\left(-\frac{\beta_k}{2} \left[\left\{ \mathbf{x}_t - \mathbf{M}_k^{-1} \mathbf{W}_k^\dagger [\mathbf{y}_t - \boldsymbol{\mu}_k] \right\}^\dagger \right. \right. \\ &\quad \left. \left. \times \mathbf{M}_k \left\{ \mathbf{x}_t - \mathbf{M}_k^{-1} \mathbf{W}_k^\dagger [\mathbf{y}_t - \boldsymbol{\mu}_k] \right\} \right] \right) \end{aligned} \quad (3.9)$$

where

$$\mathbf{M}_k := \beta_k^{-1} \mathbf{I} + \mathbf{W}_k^\dagger \mathbf{W}_k \quad (3.10)$$

Following Tipping and Bishop (1999), we can extend the EM approach of the previous section and consider the latent variables $\{\mathbf{x}_t\}$ to be missing data. If their values were known, the standard EM update equations, 2.21–2.24, 3.3, and 3.4, could be applied. Since $\{\mathbf{x}_t\}$ are actually unknown, we take the expectation value with respect to the posterior distribution, equation 3.9, denoted by angle brackets, $\langle \dots \rangle$. With the update equation

$$\hat{\boldsymbol{\mu}}_k = \frac{\sum_t \pi_k(t) \mathbf{y}_t}{\sum_t \pi_k(t)} \quad (3.11)$$

and the definition

$$\tilde{\mathbf{Y}}_k := (\mathbf{y}_{t=1} - \hat{\boldsymbol{\mu}}_k, \dots, \mathbf{y}_{t=N} - \hat{\boldsymbol{\mu}}_k), \quad (3.12)$$

the new update equations become of the following form:

$$\hat{p}_k = \frac{N_k}{N} \quad (3.13)$$

$$\langle \mathbf{G} \rangle \mathbf{\Pi}_k \tilde{\mathbf{Y}}_k^\dagger = \left(\langle \mathbf{G} \mathbf{\Pi}_k \mathbf{G}^\dagger \rangle + \frac{\alpha_k}{\beta_k} \mathbf{I} \right) \hat{\mathbf{W}}_k^\dagger \quad (3.14)$$

$$\frac{1}{\hat{\beta}_k} = \frac{1}{mN_k - \gamma_k} \sum_{t=1}^N \pi_k(t) \langle (\mathbf{y}_t - \hat{\mathbf{W}}_k \mathbf{x}_t - \boldsymbol{\mu}_k)^2 \rangle \quad (3.15)$$

$$\frac{1}{\hat{\alpha}_k} = \frac{1}{\gamma_k} \text{tr} \left[\mathbf{W}_k \mathbf{W}_k^\dagger \right] \quad (3.16)$$

There are three main differences from the standard update equations, 2.21–2.24, 3.3, and 3.4:

1. The angle brackets $\langle \dots \rangle$ indicate that an expectation value with respect to the distribution of equation 3.9 has been taken.
2. This expectation value also applies to the Hessian, from which the numbers of well-determined parameters, γ_k , are obtained. A derivation will be given in the appendix.
3. \mathbf{Y} is replaced by $\tilde{\mathbf{Y}}_k$, which includes a subtraction of the already updated parameters $\hat{\boldsymbol{\mu}}$. This, in fact, follows from a two-level EM scheme, in which the $\boldsymbol{\mu}$ -parameters are updated in a separate M-step. A discussion of the advantages of this approach is given in Tipping and Bishop (1999).

From equation 3.9 we obtain:

$$\langle \mathbf{x}_t \mathbf{x}_t^\dagger \rangle = \langle \mathbf{x}_t \rangle \langle \mathbf{x}_t^\dagger \rangle + \beta_k^{-1} \mathbf{M}_k^{-1} \quad (3.17)$$

$$\langle \mathbf{x}_t \rangle = \mathbf{M}_k^{-1} \mathbf{W}_k^\dagger (\mathbf{y}_t - \boldsymbol{\mu}_k), \quad (3.18)$$

where \mathbf{M}_k was defined in equation 3.10. Now recall that $\mathbf{g}(\mathbf{x}_t) = \mathbf{x}_t$, and apply equations 2.15, 2.16, and 3.12,

$$\langle \mathbf{G} \rangle \mathbf{\Pi}_k \tilde{\mathbf{Y}}_k^\dagger = \sum_{t=1}^N \pi_k(t) \langle \mathbf{x}_t \rangle (\mathbf{y}_t - \boldsymbol{\mu}_k)^\dagger, \quad (3.19)$$

which by inserting equation 3.18 yields:

$$\langle \mathbf{G} \rangle \mathbf{\Pi}_k \tilde{\mathbf{Y}}_k^\dagger = \sum_{t=1}^N \pi_k(t) \mathbf{M}_k^{-1} \mathbf{W}_k^\dagger (\mathbf{y}_t - \boldsymbol{\mu}_k) (\mathbf{y}_t - \boldsymbol{\mu}_k)^\dagger \quad (3.20)$$

$$= N_k \mathbf{M}_k^{-1} \mathbf{W}_k^\dagger \mathbf{S}_k, \quad (3.21)$$

where the definition of the empirical covariance matrix,

$$\mathbf{S}_k := \frac{1}{N_k} \sum_{t=1}^N \pi_k(t) (\mathbf{y}_t - \boldsymbol{\mu}_k)(\mathbf{y}_t - \boldsymbol{\mu}_k)^\dagger, \quad (3.22)$$

has been introduced (recall that $N_k = \sum_t \pi_k(t)$). From equation 3.17, we get,

$$\begin{aligned} \langle \mathbf{G} \boldsymbol{\Pi}_k \mathbf{G}^\dagger \rangle &= \sum_{t=1}^N \pi_k(t) \langle \mathbf{x}_t \mathbf{x}_t^\dagger \rangle \\ &= N_k \left[\mathbf{M}_k^{-1} \mathbf{W}_k^\dagger \mathbf{S}_k \mathbf{W}_k + \beta_k^{-1} \mathbf{I} \right] \mathbf{M}_k^{-1}, \end{aligned} \quad (3.23)$$

and inserting equations 3.23 and 3.21 into 3.14 leads to:

$$\left(\beta_k^{-1} \mathbf{I} + \mathbf{W}_k^\dagger \mathbf{S}_k \mathbf{W}_k \mathbf{M}_k^{-1} + \frac{\alpha_k}{\beta_k N_k} \mathbf{M}_k \right) \hat{\mathbf{W}}_k^\dagger = \mathbf{W}_k^\dagger \mathbf{S}_k. \quad (3.24)$$

Taking the transpose of both sides and solving for $\hat{\mathbf{W}}_k$ gives⁵

$$\hat{\mathbf{W}}_k = \mathbf{S}_k \mathbf{W}_k \left(\beta_k^{-1} \mathbf{I} + \mathbf{M}_k^{-1} \mathbf{W}_k^\dagger \mathbf{S}_k \mathbf{W}_k + \frac{\alpha_k}{\beta_k N_k} \mathbf{M}_k \right)^{-1}. \quad (3.25)$$

Note the distinction between the old parameters, \mathbf{W}_k , and the new parameters, $\hat{\mathbf{W}}_k$. Also, note that for $\alpha_k \equiv 0$, equation 3.25 reduces to equation C.14 in Tipping and Bishop (1999), so the effect of the Bayesian regularization scheme is the addition of the extra term $\frac{\alpha_k}{\beta_k N_k} \mathbf{M}_k$. Inserting equations 3.18 and 3.17 into 3.15 and applying equations 3.22 and 3.23 leads to:

$$\begin{aligned} \frac{1}{\hat{\beta}_k} &= \frac{1}{mN_k - \gamma_k} \sum_{t=1}^N \pi_k(t) \langle (\mathbf{y}_t - \hat{\mathbf{W}}_k \mathbf{x}_t - \boldsymbol{\mu}_k)^2 \rangle \\ &= \frac{1}{mN_k - \gamma_k} \sum_{t=1}^N \pi_k(t) \left\{ \|\mathbf{y}_t - \boldsymbol{\mu}_k\|^2 - 2 \langle \mathbf{x}_t^\dagger \rangle \hat{\mathbf{W}}_k^\dagger (\mathbf{y}_t - \boldsymbol{\mu}_k) \right. \\ &\quad \left. + \text{tr} \left[\hat{\mathbf{W}}_k \langle \mathbf{x}_t \mathbf{x}_t^\dagger \rangle \hat{\mathbf{W}}_k^\dagger \right] \right\} \\ &= \frac{1}{mN_k - \gamma_k} \sum_{t=1}^N \pi_k(t) \left\{ \|\mathbf{y}_t - \boldsymbol{\mu}_k\|^2 - 2 (\mathbf{y}_t - \boldsymbol{\mu}_k)^\dagger \mathbf{W}_k \mathbf{M}_k^{-1} \hat{\mathbf{W}}_k^\dagger (\mathbf{y}_t - \boldsymbol{\mu}_k) \right\} \\ &\quad + N_k \text{tr} \left[\hat{\mathbf{W}}_k (\mathbf{M}_k^{-1} \mathbf{W}_k^\dagger \mathbf{S}_k \mathbf{W}_k + \beta_k^{-1} \mathbf{I}) \mathbf{M}_k^{-1} \hat{\mathbf{W}}_k^\dagger \right] \end{aligned}$$

⁵ The following expression is derived to get the update equation into a form equivalent to that stated in Tipping and Bishop (1999). For a practical implementation, the matrix inversion would not be carried out explicitly.

$$= \frac{N_k}{mN_k - \gamma_k} \text{tr} \left[\mathbf{S}_k - 2\mathbf{S}_k\mathbf{W}_k\mathbf{M}_k^{-1}\hat{\mathbf{W}}_k^\dagger + \hat{\mathbf{W}}_k(\mathbf{M}_k^{-1}\mathbf{W}_k^\dagger\mathbf{S}_k\mathbf{W}_k + \beta_k^{-1}\mathbf{I})\mathbf{M}_k^{-1}\hat{\mathbf{W}}_k^\dagger \right].$$

By making use of equation 3.25, this gives

$$\frac{1}{\hat{\beta}_k} = \frac{N_k}{mN_k - \gamma_k} \text{tr} \left[\mathbf{S}_k - \mathbf{S}_k\mathbf{W}_k\mathbf{M}_k^{-1}\hat{\mathbf{W}}_k^\dagger - \frac{\alpha_k}{N_k\beta_k}\hat{\mathbf{W}}_k\hat{\mathbf{W}}_k^\dagger \right] \quad (3.26)$$

where kernels with $N_k \leq \frac{\gamma_k}{m}$ are pruned. For $\alpha_k \equiv 0$ and $\gamma_k \equiv 0$ (that is, the unregularized case), this reduces to equation C.15 in Tipping and Bishop (1999).

4 Experiments

4.1 Unconditional Probability Densities. When conditioning the data vectors \mathbf{y}_t on a constant element $\mathbf{g}(\mathbf{x}_t) \equiv 1$, equation 1.2 reduces to the modeling of unconditional probability densities, $P(\mathbf{y}_t|\mathbf{x}_t) = P(\mathbf{y}_t|1)$, where the weights \mathbf{w}_k exiting the constant unit are equivalent to the centers of the gaussian kernels. The prior on the weights is slightly modified from the form of equation 2.1 in that it is centered on the mean of the data rather than the origin.⁶ This leads to a small modification of the update equations, 2.21–2.24, as discussed in the appendix (equations A.38 and A.39). This study focuses on binary classification problems, where the class conditional distributions $P(\mathbf{y}_t|class = 1)$ and $P(\mathbf{y}_t|class = 2)$ are modeled separately with two different networks. The generalization errors of these separate density estimates are measured in terms of the negative normalized log-likelihood for the test data,

$$E_c := -\frac{1}{N_c} \sum_{\mathbf{y}_t \in D_{test}^c} \ln P(\mathbf{y}_t|class = c), \quad (4.1)$$

where D_{test}^c is an independent test set of cardinality N_c for class c . The networks are then combined into a modular structure to predict the probability for a given class, conditional on the input vector \mathbf{y}_t , by Bayes' rule:

$$P(class = c|\mathbf{y}_t) = \frac{P(\mathbf{y}_t|class = c)P(class = c)}{P(\mathbf{y}_t)}, \quad (4.2)$$

where $P(class = 1) + P(class = 2) = 1$, and $P(\mathbf{y}_t) = \sum_{c=1}^2 P(\mathbf{y}_t|class = c)P(class = c)$. The further adjustable parameter $P(class = 1)$ represents the

⁶ This is equivalent to a force that pulls the kernel centers toward the mean of the distribution, whereby kernels are discouraged from homing in on remote outliers.

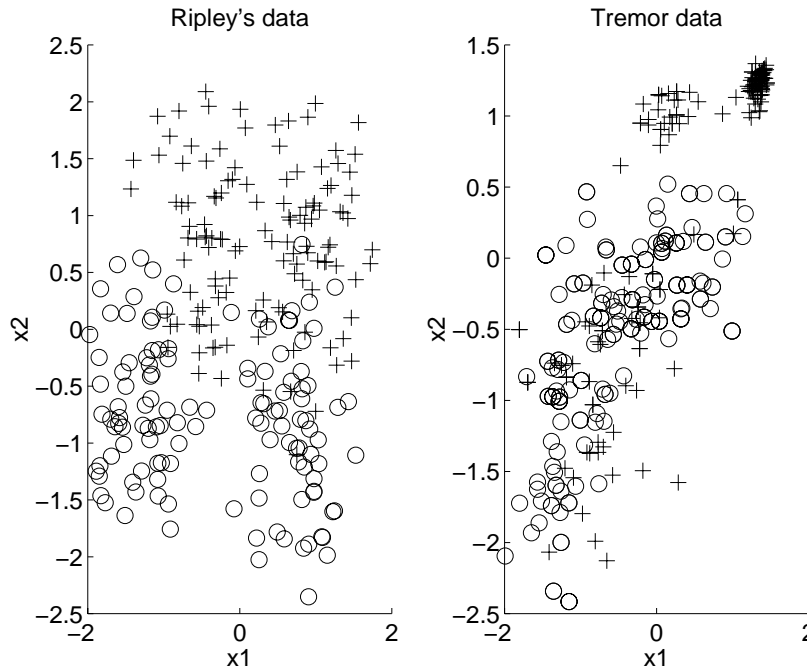


Figure 2: Two classification problems chosen for the numerical experiments. (Left) Ripley's synthetic data. (Right) Tremor data.

prior for class 1 and can (if no actual prior knowledge is available) easily be optimized so as to minimize the misclassification on the training set. A straightforward approach is to follow a hard classification scheme and assign exemplar \mathbf{y}_t to class c if $P(\text{class} = c | \mathbf{y}_t) > 0.5$. The generalization performance can then be measured on an independent test set in terms of the percentage misclassification (E_{class}).

The Bayesian regularization scheme was tested on the following three data sets. *Ripley* is a synthetic data set taken from Ripley (1994). There are two features and two classes, where each class has a bimodal distribution, as seen from Figure 2 (left). The class distributions are given by equal mixtures of two normal distributions, whose overlap is chosen to allow a best-possible error rate (Bayes limit) of about 8%. The networks were trained on $N_{\text{train}} = 250$ training exemplars (125 for each class) and tested on an independent set of size $N_{\text{test}} = 1000$.

Tremor is a real-world data set, plotted in Figure 2 (right), where the objective is to identify Parkinson's disease on the basis of muscle tremor. The data set, collected by Spyers-Ashby (1996), consists of two input fea-

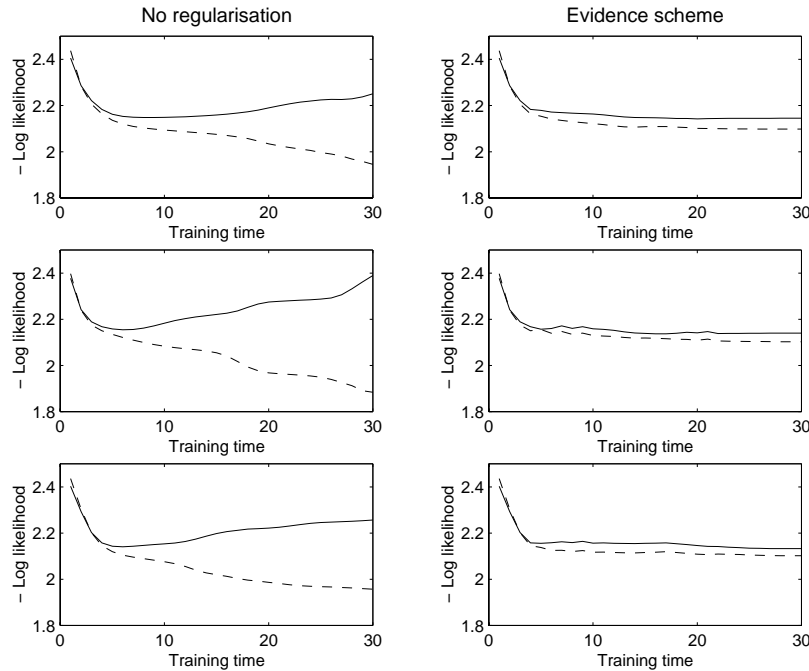


Figure 3: Evolution of E_1 , the normalized negative log-likelihood for class 1, when training a GM network with $K = 10$ kernels on Ripley's data set. Dashed lines: Training set performance; solid lines: Test set performance. (Left) Results of unregularized training. (Right) Networks regularized with the evidence scheme. The three rows refer to different initializations of the parameters. Abscissa: training time; ordinate: E_1 .

tures derived from measurements of arm muscle tremor and a class label representing patient or nonpatient: $N_{train} = 179$, $N_{test} = 178$.

The task of the *Kaposi* problem is to predict whether an AIDS patient is likely to contract Kaposi's sarcoma, a vascular tumor that is often aggressive in patients with underlying immunosuppression. The classification is done on the basis of four immunological input variables measuring the concentrations of various lymphocytes, $N_{train} = 263$, $N_{test} = 39$, with two different partitions into training and test sets.

Figure 3 shows the evolution of E_1 , the normalized negative test set log-likelihood for *class 1* (see equation 4.1), for training three differently initialized networks (all with $K = 10$ kernels) on Ripley's data set. Without

regularization⁷ (left column) we observe the typical behavior of overfitting. While E_1 naturally decreases on the training set (dashed line), the performance on the test set (solid line) deteriorates from a certain (unknown) optimum number of adaptation steps on. This deficiency is significantly improved when the evidence method for regularization is applied (right column). In this case, E_1 on the test set reaches a constant plateau and does not vary much in response to changes in the training time, thus considerably reducing the risk of overfitting.

The simulations were then repeated 24 times for four initial kernel numbers ($K = 5, 10, 15, 20$), two initial kernel widths ($\sigma_0 = 1/\sqrt{\beta_0} = 0.5, 1$), and three initializations of the kernel centroids,⁸ where in each case training was carried out over a fixed number of $T = 20$ adaptation steps. (The output weights were always initialized uniformly: $p_k = 1/K \forall k$.) The left column of Figure 4 shows three scatterplots (for E_1 , top, E_2 , middle, and E_{class} , bottom), where for corresponding simulations (starting from the same initialization) results obtained without regularization (abscissa) are plotted against those obtained with the Bayesian regularization scheme (ordinate). It is clearly seen that unregularized training leads to large variations in both performance measures (E_1 and E_2 indicating the accuracy of modeling the class-conditional distributions, and E_{class} showing the classification error), with large kernel numbers K usually causing drastic overfitting. This is effectively prevented with the Bayesian regularization scheme, which results in a rather constant performance irrespective of the model complexity K and therefore considerably reduces the need for any sophisticated model selection methods. Also, note that the achieved classification performance with regularization is always close to the Bayes limit.

Similar simulations were carried out on the other two data sets.⁹ For the Tremor data, the results are shown on the right of Figure 4, which shows the same scatter diagrams as for Ripley's data. Again, it is seen that the evidence scheme leads to a considerable stabilization of the generalization performance with respect to changes in the model complexity, although unregularized training occasionally results in a better modeling of the probability density of class 1 (top right) and, consequently, a better classification performance (bottom right).

In order to test the statistical significance of the improvement achieved with the evidence method, a matched-pairs t -test was carried out (as de-

⁷ Recall that training without regularization is equivalent to equations 2.21–2.24 with $\alpha_k \equiv 0$, $\gamma_{ki} \equiv 0$.

⁸ The $\{\mu_k\}$ were drawn from $N(\mathcal{M}_c, 1)$, where \mathcal{M}_c is the unconditional mean for the c th class.

⁹ For the Tremor data, only two rather than three different initializations of the kernel centers μ_k were chosen (leading to 16 simulations), whereas the simulations on the Kaposi data were repeated for two different partitions of the data into training and test sets (giving 48 simulations).

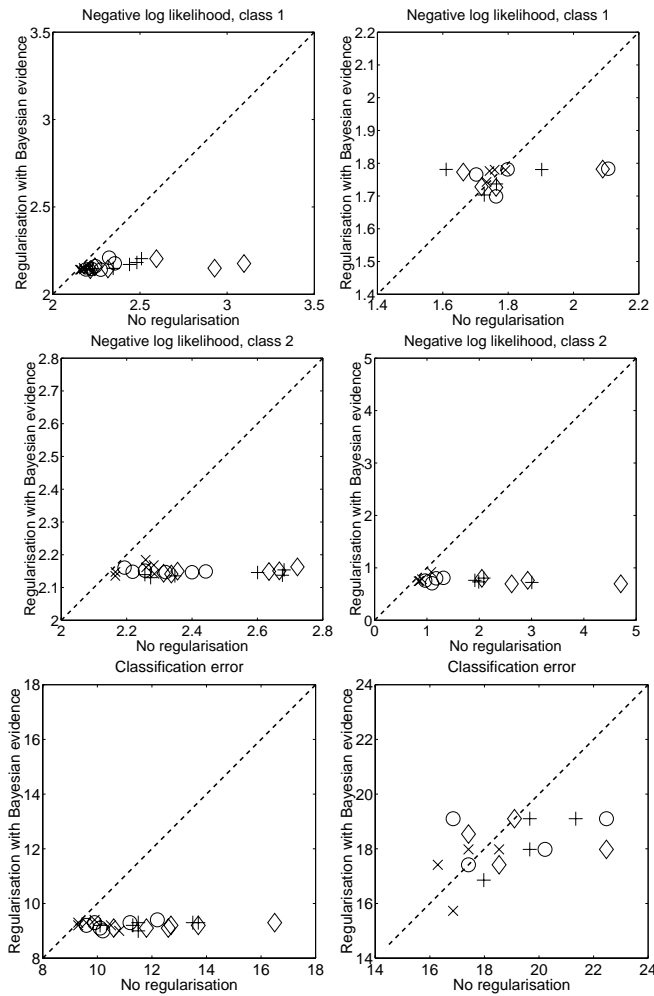


Figure 4: Comparison between unregularized training (abscissa) and the evidence scheme (ordinate), applied to Ripley's synthetic data (left) and the Tremor data (right). (Top) E_1 (normalized negative log-likelihood for class 1, test set). (Middle) E_2 (normalized negative log-likelihood for class 2, test set). (Bottom) E_{class} (percentage misclassification, test set). The symbols refer to different numbers of kernels in the network. X-marks: $K = 5$; circles: $K = 10$; crosses: $K = 15$; diamonds: $K = 20$. The dashed line indicates equal performance. Symbols below that line point to a performance improvement as a result of regularization with the evidence method.

Table 1: Matched Pairs t-Test for a Comparison Between the Evidence Scheme and Unregularized Training over a Fixed Number of $T = 20$ Training Steps.

Data	Measure	t -Statistic	Critical Value	Evidence Method
Ripley	E_1	4.02	± 1.71	Better
	E_2	6.24	± 1.71	Better
	E_{class}	5.97	± 1.71	Better
Tremor	E_1	0.99	± 1.75	Equal
	E_2	3.97	± 1.75	Better
	E_{class}	1.94	± 1.75	Better
Kaposi	E_1	8.84	± 1.68	Better
	E_2	5.90	± 1.68	Better
	E_{class}	2.34	± 1.68	Better

Notes: E_1 and E_2 represent the negative normalized test set log-likelihoods for the two classes (see equation 4.1) and are a measure of how good the class-conditional distributions are modeled. E_{class} is the percentage misclassification. Positive values indicate that the evidence method leads to a better performance; for negative values, the alternative method is superior. The deviation is significant (at a 95% significance level) if the modulus of the statistic is larger than the critical value.

scribed, e.g., in Hoel, 1984). For corresponding simulations, the differences in the performance measures E_1 , E_2 , and E_{class} between regularized and unregularized training were calculated.¹⁰ Positive values indicate that the evidence method leads to a better performance; for negative values, the alternative method is superior. The results are listed in Table 1, together with the critical value¹¹ for rejecting the null hypothesis of equal performance. The Bayesian regularization scheme consistently leads to an improvement in all three performance measures, which, except for E_1 of the Tremor data, is always significant.

In a further study, the results of the evidence approach were compared with the best test set performance obtained with the unregularized EM algorithm, that is, with the minimum of test set learning curves like those of Figure 3 (left). This is akin to the method of early stopping, except that the selection is done on the basis of the test set rather than a separate valida-

¹⁰ In more detail, let z_i denote the results obtained without regularization and z'_i the results obtained with regularization. Now consider the differences $\delta_i := z_i - z'_i$ and calculate the empirical variance $S^2 := \frac{1}{n-1} \sum_{i=1}^n (\delta_i - \bar{\delta})^2$, in which $\bar{\delta}$ denotes the empirical mean: $\bar{\delta} := \frac{1}{n} \sum_{i=1}^n \delta_i$. The t -statistic can now be calculated according to $t = \frac{\sqrt{n} \bar{\delta}}{S}$, which is compared with the critical value for $(n-1)$ degrees of freedom and a given (in this case, 95%) significance level.

¹¹ The critical value depends on the number of degrees of freedom, which is the number of different simulations minus one (Ripley: 23, Tremor: 15, Kaposi: 47). See, for instance, Hoel (1984).

Table 2: Matched Pairs t-Test for a Comparison Between the Evidence Scheme and Optimal Early Stopping.

Data	Measure	t -Statistic	Critical Value	Evidence Method
Ripley	E_1	3.40	± 1.71	Better
	E_2	2.74	± 1.71	Better
	E_{class}	3.66	± 1.71	Better
Tremor	E_1	-1.97	± 1.75	Worse
	E_2	-4.31	± 1.75	Worse
	E_{class}	-2.33	± 1.75	Worse
Kaposi	E_1	3.02	± 1.68	Better
	E_2	-1.46	± 1.68	Equal
	E_{class}	-1.62	± 1.68	Equal

Note: For details, see the caption of Figure 1.

tion set. Since the latter would reduce the number of training exemplars and, consequently, degrade the performance of the prediction model, the alternative results are better than what could be obtained in real applications.

The results are listed in Table 2. On Ripley's data, the evidence method achieves a consistent improvement in terms of all three performance measures, and for the Kaposi data, the modeling of one of the class-conditional probability distributions is significantly improved. However, on the Tremor data, the evidence method turns out to be inferior to the alternative approaches. A possible explanation for this deficiency will be given in section 5.

4.2 Conditional Probability Densities. The objective of the second part of this study is the prediction of the conditional probability density of a noisy time series, for which the synthetic benchmark problem of Husmeier and Taylor (1997) was chosen. Two stochastic dynamical systems are coupled stochastically according to

$$y_{t+1} = \Theta(\xi_t - \theta) [\alpha_t y_t (1 - y_t)] + [1 - \Theta(\xi_t - \theta)] [1 - y_t^{\kappa_t}], \quad (4.3)$$

where $\alpha_t \in [3, 4]$, $\kappa_t \in [0.5, 1.25]$, and $\xi_t \in [0, 1]$ are random variables uniformly distributed in the respective interval, $\theta = 1/3$, and $\Theta(\cdot)$ denotes the Heaviside function. The resulting time series is a first-order Markov process with a bimodal conditional probability distribution in state-space. The networks were trained on a time-series segment of length $N_{train} = 200$, and the generalization performance was tested on $N_{test} = 1000$ independent exemplars. The chosen network architecture was of the form depicted in Figure 1, where the weights on connections between the input and the first hidden layer were fixed at randomly selected values. Note that this corresponds

to a random vector functional link network, whose universal approximator power is discussed in Igelnik and Pao (1995).

Training with the Bayesian evidence scheme followed equations 2.21–2.24. Recall that this reduces to the unregularized EM algorithm for $\alpha_k \equiv 0$ and $\gamma_{ki} \equiv 0$. The following alternative regularization schemes were employed for a comparison:

Eigenvalue (EV) cutoff. This is equivalent to the unregularized EM algorithm except that when updating the weights \mathbf{w}_{ki} according to equation 2.22 (with $\alpha_k = 0$), the matrix on the left, $\mathbf{G}\mathbf{\Pi}_k\mathbf{G}^\dagger$, is singular-value decomposed, and small eigenvalues¹² are set to infinity. When solving for \mathbf{w}_{ki} , the contributions along the corresponding eigenvalues thus disappear. This is equivalent to a projection of the full solution vector onto a subspace perpendicular to the domain spanned by the eigenvectors belonging to small eigenvalues or, put differently, components of \mathbf{w}_{ki} that are poorly determined by the data are discarded. (For a detailed exposition of this method, see Press, Teukolsky, Vetterling, & Flannery, 1992, chap. 2).

Weight decay. The weights \mathbf{w}_{ki} are updated according to equation 2.22 with fixed values for α_k . This is equivalent to a simple weight decay scheme. Two different values of the weight-decay hyperparameters were used: $\alpha_k = 0.01$ and $\alpha_k = 0.1$ (the same for all kernels k).

Naive Bayes. The kernel variances are updated according to

$$\frac{1}{\hat{\beta}_{ki}} = \frac{\sum_t \pi_k(t) [y_{ti} - f_{ki}]^2 + 2\rho}{\sum_t \pi_k(t)} \geq \frac{2\rho}{N}, \quad (4.4)$$

where the hyperparameter ρ is set to the value that gave the best generalization performance in Ormoneit and Tresp (1996): $\rho = 0.1$. Effectively this imposes a lower bound on $1/\beta_{ki}$. The adaptation of the weights \mathbf{w}_{ki} follows from equation 2.22, with $\alpha_k \equiv 0.1 \forall k$. Note that this is equivalent to a Bayesian maximum a posteriori approach with a gamma prior on the inverse variances β_{ki} and a gaussian prior on the weights \mathbf{w}_{ki} .

For each method, an ensemble of 80 networks was created in the following way. The random weights in the network were drawn from four different gaussian distributions $N(0, \sigma_{rand})$ with $\ln \sigma_{rand} \in \{-1, 0, 1, 2\}$. For each of the four distribution widths σ_{rand} , 20 weight configurations were drawn, based on different random number generator seeds. Each set of net-

¹² In this study, an eigenvalue was defined as small when $\varepsilon_v < 10^{-6} \varepsilon_{\max}$, where ε_{\max} is the maximum eigenvalue in that weight group. The same cutoff value is recommended in Press et al. (1992).

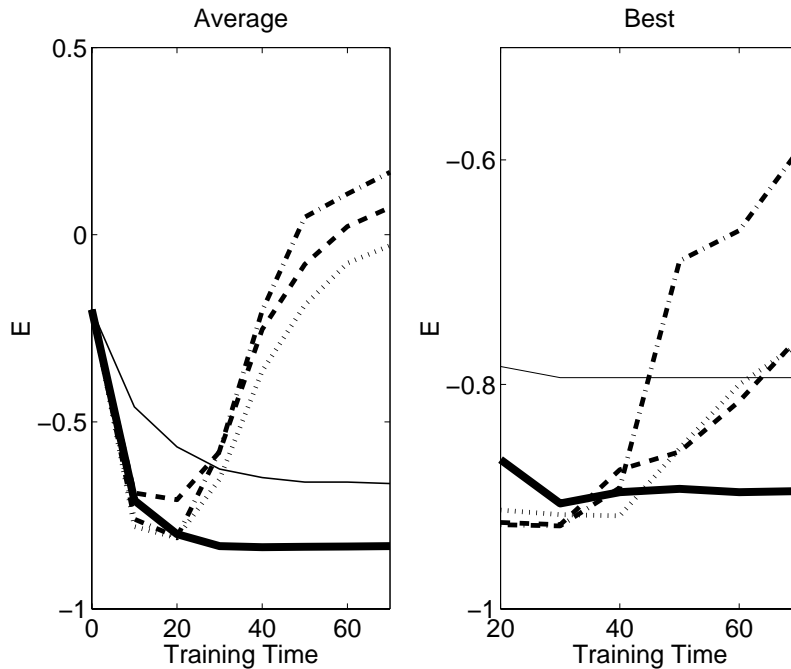


Figure 5: Dependence of the generalization performance on the regularization method and the training time for the time-series prediction problem. (Left) Evolution of the average generalization “error,” measured in terms of the mean negative normalized test-set log-likelihood E_{test} . (Right) Evolution of the best generalization “error,” measured in terms of the minimum of E_{test} . The abscissa represents the number of adaptation steps. Five regularization methods were tested: dotted line: EV cutoff; dashed-dotted line: weight decay, $\alpha = 0.01$; dashed line: weight decay, $\alpha = 0.1$; thin solid line: naive Bayes; thick solid line: Bayesian evidence method.

works with identical σ_{rand} was further subdivided into four subsets, which differed with respect to the initialization of the adaptable weights w_{ki} , drawn from (1) $N(0, 0.1)$, (2) $N(0, 0.25)$, (3) $N(0, 0.5)$, or (4) $N(0, 1.0)$. The remaining parameters, β_k and p_k , were initialized as before: $p_k = 1/K \forall k$, $\beta_k = 1 \forall k$. Each network contained 10 tanh units in the first hidden layer and $K = 10$ gaussian kernels in the second hidden layer.

Figure 5 (left) shows the dependence of the average generalization performance on the regularization method and the training time. The graphs represent the mean of the negative normalized test set log-likelihood, $E_{test} = -\frac{1}{N_{test}} \sum_{t=1}^{N_{test}} \ln P(y_{t+1}|y_t)$. The best results are obtained with the evidence scheme, but the difference is not statistically significant. A clear improve-

ment, however, is achieved with respect to variations in the length of training time. The simulations suggest that no overfitting occurs for the evidence scheme. Compare this with the first three regularization methods (EV cutoff, weight decay with $\alpha = 0.01$, weight decay with $\alpha = 0.1$), which show strong overfitting after about 20 to 30 adaptation steps. This calls for model selection by cross-validation, which, however, would reduce the amount of training data and is therefore likely to incur a loss of modeling accuracy. The naive Bayesian method, where the hyperparameter ρ was set to the optimal value in Ormoneit and Tresp (1996), is strongly overregularized, with an increase of the mean of E_{test} by about 1.7 standard deviations.

Figure 5 (right) shows the dependence of the best generalization performance on the regularization method and the training time, that is, the graphs represent the minimum of E_{test} . Again, the evidence scheme leads to a considerable stabilization of the training process with respect to changes in the training time, whereas the first three methods (EV cutoff, $\alpha = 0.01$, $\alpha = 0.1$) show drastic overfitting. For small training times, the best under-regularized models are slightly better than the best model obtained with the evidence scheme, but this can, in general, not be exploited in practice (since model selection requires a cross-validation set, thereby incurring the problems discussed before). The naive Bayesian method turns out to be strongly overregularized again.

4.3 Latent Space Model (MPPCA). The last application is taken from sleep research, where the objective is to distinguish different sleep phases. The data consisted of two classes, deep sleep and rapid eye movement, where each class contained $N = 960$ 10-dimensional feature vectors.¹³ The data were randomly split into two sets of equal size for training and testing. An MPPCA model of latent dimension $n = 2$ was trained over a fixed number of 60 adaptation steps. For the Bayesian evidence scheme, the weights \mathbf{w}_k and kernel precisions β_k were updated according to equations 3.25 and 3.26. Unregularized training followed the same update rules, but set $\alpha_k \equiv 0$ and $\gamma_k \equiv 0$. (Note that this is equivalent to equations C.14 and C.15 in Tipping & Bishop, 1999.)

The top graphs of Figure 6 show a typical evolution of the training process. When training with the unregularized maximum likelihood scheme, the negative log-likelihood $E = -\frac{1}{N} \ln P(\mathbf{D}|\text{class})$ continually decreases on the training set (narrow dashed lines), but soon increases on the test set (bold dashed lines) as a result of overfitting. When training with the Bayesian evidence method, the training set performance (narrow solid lines) becomes (naturally) worse, but the test set performance (bold solid lines) is improved, and overfitting is effectively prevented.

¹³ These vectors were obtained by fitting a tenth-order autoregressive model to a moving fixed-size window of an EEG signal.

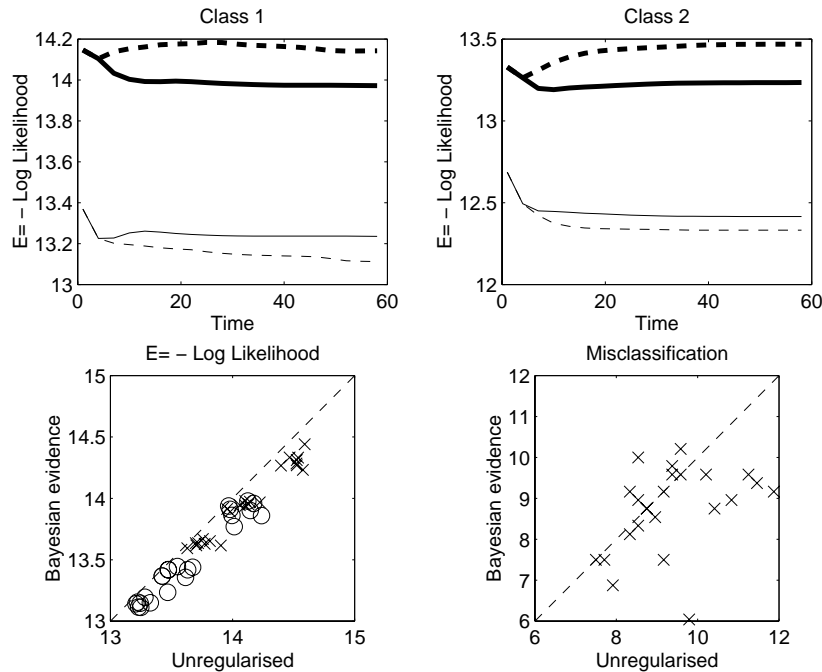


Figure 6: The Bayesian evidence scheme applied to MPPCA (mixture of probabilistic principal component analyzers). The top row shows the evolution of E , the normalized negative log-likelihood, on the training set (narrow lines) and the test set (bold lines) without regularization (dashed lines) and with the Bayesian evidence scheme (solid lines). The scatterplots in the bottom row compare the Bayesian evidence scheme with unregularized training for a total of 24 simulations, where the figure on the left shows the negative test set log-likelihood for classes 1 (crosses) and 2 (circles) and the figure on the right shows the percentage misclassification scores. The diagonal dashed lines indicate equal performance; symbols below the dashed line point to a performance improvement as a result of applying the Bayesian evidence scheme. Note the small but consistent improvement in the plot on the bottom left. The respective t -statistics are shown in Table 3.

To test the statistical significance of the results, the simulations were repeated 24 times for two kernel numbers ($K = 4, 6$), three different partitions of the data into training and test sets, and four different initializations.¹⁴

¹⁴ Two initial kernel widths were chosen— $\beta_k^{-1} = 0.5$ and $\beta_k^{-1} = 1.0$ —and the simulations started from two different random number generator seeds. The initial weights were set to $w_{ki} = 0$, and the components of the kernel centers, μ_k , were drawn from a normal distribution centered at the mean of the respective class. The first five adaptation steps

Table 3: Mixture of Probabilistic Principal Component Analyzers.

	Unregularized (Maximum Likelihood)	Optimal Early Stopping
$E_1 = -\ln P(\mathbf{D} \text{class} = 1)$	9.65	4.58
$E_2 = -\ln P(\mathbf{D} \text{class} = 2)$	7.93	1.40
Misclassification	2.36	0.74

Notes: The table shows the t -statistics for a matched pairs test with 23 degrees of freedom and a critical value of 1.71. Positive values indicate that the Bayesian evidence scheme gives better results; boldface figures indicate that this improvement is significant (at a 95% significance level). Left column: Comparison between the Bayesian evidence scheme and unregularized training. Right column: Comparison between the Bayesian evidence scheme and optimal early stopping (described in the text).

The results are plotted in the bottom row of Figure 6, which shows scatter diagrams similar to those discussed in section 4.1.

The figure on the bottom left shows a comparison between unregularized (maximum likelihood) training and the Bayesian evidence scheme for modeling the two class-conditional distributions. Note that the dashed diagonal line indicates equal performance of the two methods, whereas symbols below the dashed line point to a performance improvement as a result of applying the evidence scheme. This is, in fact, observed in the figure, which suggests that although the improvement achieved with the evidence approach is relatively small, it is consistent in that it occurred in every simulation. A matched-pairs t -test gives a value of 9.65 (see Table 3), which is larger than the critical value of 1.71 (for a 95% significance level) and therefore suggests that the improvement is significant. Similarly, the figure on the bottom right shows a comparison between unregularized training and the evidence scheme with respect to the actual classification scores. The respective t -statistic, shown in Table 3, is 2.36; hence the improvement achieved with Bayesian regularization is less dramatic but still statistically significant.

Finally, the evidence scheme was also compared with the method of optimal early stopping. (Recall that this is an idealized version of early stopping that gives better results than can be achieved in practice.) The results of a matched-pairs t -test are given in Table 3, which suggests that the evidence scheme still gives better results, although this is less pronounced and statistically significant for only one (out of three) performance measure.

were unregularized and followed the (faster) direct update scheme of equations 3.12 and 3.13 in Tipping and Bishop (1999).

5 Discussion

This study has applied the Bayesian evidence scheme to the regularization of probability-density estimating neural networks. This is to be distinguished from Roberts, Husmeier, Rezek, and Penny (1998), where the evidence scheme was applied to model selection, whereas the actual training process was unregularized. Also, note that in Roberts et al. (1998), only the lowest-order approximation to the Hessian, equivalent to the first term in the expansion of equations A.28 and A.37 was applied.

The numerical experiments suggest that the evidence method derived in this study leads to a considerable stabilization of the generalization performance with respect to variations in the model complexity (see Figure 4) and training time (see Figures 3 and 5). A comparison with alternative regularizers gave, in general, very favorable results. The following restrictions, however, are to be noted.

When modeling unconditional probability densities according to section 4.1, a gaussian prior is introduced on the kernel centers $\mathbf{f}_k = \mathbf{w}_k$ themselves (rather than on parameters determining the functional mapping onto the kernel centers). This explains the comparatively poor performance on the tremor data, where one of the class-conditional distributions is strongly skewed and considerably deviates from a gaussian. Moreover, for classification problems per se, a gaussian prior might be suboptimal, since one would prefer to deploy more kernels near the classification boundary rather than in the interior of the distribution. Consequently, although the modeling of the actual distribution tends to be improved by the evidence scheme, this is not necessarily reflected in the classification performance (as seen for the Kaposi data; see Table 2).

These restrictions do not apply to the modeling of conditional densities, for which the kernel centers are modeled by generalized linear functions according to equation 1.1, and the chosen form of the prior is equivalent to the standard complexity regularization of linear weight decay. Modeling the kernel centers as generalized linear functions implies that for a given approximation accuracy, the model complexity will usually be larger than that of an equivalent one-hidden-layer neural network. However, Igel'nik and Pao (1995) claim that for an appropriate stochastic choice of the basis functions (the functions $\mathbf{g}(\mathbf{x}_t)$ in equation 1.1), this increase in the complexity may not be dramatic.

Probabilistic principal component analysis is based on a generalized linear model for predicting conditional densities $P(\mathbf{y}|\mathbf{x})$, but in fact integrates the latent variables \mathbf{x} out so as to model unconditional densities. This is done according to the EM algorithm, which requires taking the expectation value of the Hessian with respect to the posterior distribution $P(\mathbf{x}|\mathbf{y}, k)$ of equation 3.9. In the work presented here, this has been done only for the lowest-order approximation to the Hessian, as discussed in the appendix. The resulting update algorithm was found to achieve a significant improvement

over training with the maximum likelihood method for both the modeling of the class-conditional distributions and the classification performance. On a comparison with an idealized form of early stopping, the evidence method still tended to achieve better results, although this was less pronounced and not always statistically significant. The derivation of the expectation value of a higher-order approximation to the Hessian, which takes the second term in equation A.28 into account, is the subject of future research.

6 Conclusion

Training probability-density estimating neural networks with the EM algorithm aims to maximize the likelihood of the training set and therefore leads to severe overfitting for sparse data. The proposed regularization method adopts the Bayesian evidence scheme, whereby the hyperparameters of the prior are optimized by type II maximum likelihood, that is, after marginalizing over the parameters. This is done by Laplace approximation, which requires the derivation of the Hessian of the log-likelihood function. The incorporation of this approach into the standard training scheme leads to a modified form of the EM algorithm, which includes extra regularization terms whose hyperparameters are adapted on-line after each EM cycle. The method has been applied to the direct modeling of unconditional densities, the indirect modeling of unconditional densities via a latent-space model, and the prediction of conditional probability densities. Simulations on four classification problems and one time-series prediction task suggest that the generalization performance is significantly improved over unregularized maximum likelihood training and that the evidence scheme tends to outperform the alternative regularization methods employed in this study. Moreover, most simulation results suggest that the training process is stabilized with respect to changes in the training time and the model complexity, which prevents overfitting and greatly reduces the need for model selection.

Appendix

A.1 Derivation of the Regularized EM Algorithm. Consider a density-estimating network with parameters $\mathbf{q} = \{\mathbf{w}, \mathbf{p}, \beta\}$, and define the K by N matrix of binary variables $\lambda_k(t)$,

$$\Lambda := (\lambda(1), \dots, \lambda(N)), \quad \lambda(t) := \begin{pmatrix} \lambda_1(t) \\ \vdots \\ \lambda_K(t) \end{pmatrix}, \quad \lambda_k(t) \in \{0, 1\},$$

$$\|\lambda(t)\| = 1 \quad \forall t, \tag{A.1}$$

where $\lambda_k(t) = 1$ indicates that the t th exemplar \mathbf{y}_t has been generated from the k th component of the mixture distribution, equation 1.2, and the last

equation on the right implies that at any particular time t , one and only one $\lambda_k(t)$ is set to 1. Moreover, define

$$\Psi(\mathbf{q}, \Lambda) := -\ln P(\mathbf{D}, \Lambda | \mathbf{q}) \quad (\text{A.2})$$

$$U(\mathbf{q} | \mathbf{q}') := \langle \Psi(\mathbf{q}, \Lambda) \rangle_{\Lambda | \mathbf{D}, \mathbf{q}'} \quad (\text{A.3})$$

$$S(\mathbf{q} | \mathbf{q}') := \langle \ln P(\Lambda | \mathbf{D}, \mathbf{q}) \rangle_{\Lambda | \mathbf{D}, \mathbf{q}'}, \quad (\text{A.4})$$

where the abbreviation $\langle f \rangle_{\Lambda | \mathbf{D}, \mathbf{q}'} := \int f(\Lambda) P(\Lambda | \mathbf{D}, \mathbf{q}') d\Lambda$ has been applied. Note the distinction between \mathbf{q} and \mathbf{q}' , where the latter denote the current or old parameters, which are kept fixed, and the former the new parameters, which are adapted (see below). From definition 2.3 and equations A.2 through A.4 we obtain:

$$\begin{aligned} E_o(\mathbf{q}) &= -\ln P(\mathbf{D} | \mathbf{q}) = \ln P(\Lambda | \mathbf{D}, \mathbf{q}) - \ln P(\mathbf{D}, \Lambda | \mathbf{q}) \\ &= U(\mathbf{q} | \mathbf{q}') + S(\mathbf{q} | \mathbf{q}') \end{aligned} \quad (\text{A.5})$$

$$[\nabla E_o(\mathbf{q})]_{\mathbf{q}=\mathbf{q}'} = [\nabla U(\mathbf{q} | \mathbf{q}')]_{\mathbf{q}=\mathbf{q}'}, \quad (\text{A.6})$$

where equation A.6 follows from the fact that $S(\mathbf{q} | \mathbf{q}')$ has its maximum at $\mathbf{q} = \mathbf{q}'$ (e.g., Papoulis, 1991). The joint probability for the class labels and the data is given by¹⁵

$$P(\mathbf{D}, \Lambda | \mathbf{q}) = \prod_{t=1}^N \prod_{k=1}^K (p_k P(y_t | \mathbf{q}, k))^{\lambda_k(t)} \quad (\text{A.7})$$

$$= \prod_{t=1}^N \prod_{k=1}^K \prod_{i=1}^m \left[p_k \sqrt{\frac{\beta_{ki}}{2\pi}} \exp\left(-\frac{\beta_{ki}}{2} (y_{ti} - f_{ki})^2\right) \right]^{\lambda_k(t)} \quad (\text{A.8})$$

and, with equation A.2,

$$\Psi(\mathbf{q}, \Lambda) = \sum_{t=1}^N \sum_{k=1}^K \sum_{i=1}^m \lambda_k(t) \left[\frac{\beta_{ki}}{2} (y_{ti} - f_{ki})^2 - \ln p_k - \frac{1}{2} \ln \left(\frac{\beta_{ki}}{2\pi} \right) \right] \quad (\text{A.9})$$

Since $\lambda_k(t)$ is binary and thus $\langle \lambda_k(t) \rangle_{\Lambda | \mathbf{D}, \mathbf{q}'} = P(\lambda_k(t) = 1 | \mathbf{D}, \mathbf{q}') = \pi_k(t)$, equations A.3 and A.9 lead to

$$U(\mathbf{q} | \mathbf{q}') = \sum_{t=1}^N \sum_{k=1}^K \sum_{i=1}^m \pi_k(t) \left[\frac{\beta_{ki}}{2} (y_{ti} - f_{ki})^2 - \ln p_k - \frac{1}{2} \ln \left(\frac{\beta_{ki}}{2\pi} \right) \right]. \quad (\text{A.10})$$

¹⁵ See, for instance, Bishop (1995), Dempster et al. (1977), and Husmeier (1998).

Taking the derivatives of U gives¹⁶

$$\frac{\partial U}{\partial \alpha_k} = 0 \quad (\text{A.11})$$

$$\frac{\partial U}{\partial \beta_{ki}} = \sum_{t=1}^N \frac{\pi_k(t)}{2} \left[(y_{ti} - f_{ki})^2 - \frac{1}{\beta_{ki}} \right] \quad (\text{A.12})$$

$$\frac{\partial U}{\partial p_k} = -\frac{m}{p_k} \sum_{t=1}^N \pi_k(t) + \frac{m}{p_K} \sum_{t=1}^N \pi_K(t) \quad (\text{A.13})$$

$$\begin{aligned} -\nabla_{\mathbf{w}_{ki}} U &= \beta_{ki} \sum_{t=1}^N \pi_k(t) [y_{ti} - f(\mathbf{x}_t; \mathbf{w}_{ki})] \nabla_{\mathbf{w}_{ki}} f(\mathbf{x}_t; \mathbf{w}_{ki}) \\ &= \beta_{ki} \left(\sum_{t=1}^N \pi_k(t) y_{ti} \mathbf{g}(\mathbf{x}_t) - \sum_{t=1}^N \pi_k(t) \mathbf{g}(\mathbf{x}_t) [\mathbf{g}(\mathbf{x}_t)]^\dagger \mathbf{w}_{ki} \right) \\ &= \beta_{ki} \left[\mathbf{G} \Pi_k \mathbf{y}_{\cdot i} - \left(\mathbf{G} \Pi_k \mathbf{G}^\dagger \right) \mathbf{w}_{ki} \right], \end{aligned} \quad (\text{A.14})$$

where in the last step, the definitions 2.15, 2.16, and 2.19 have been applied. On inserting the expression for the prior on \mathbf{w} , equation 2.1, into the definition of the regularization term, equation 2.4, we obtain

$$R(\mathbf{w}; \alpha) = \sum_{k=1}^K \sum_{i=1}^m \frac{\alpha_k}{2} \mathbf{w}_{ki}^\dagger \mathbf{w}_{ki} - \frac{m\tilde{m}}{2} \sum_{k=1}^K \ln \frac{\alpha_k}{2\pi}, \quad (\text{A.15})$$

and for the derivatives,

$$\nabla_{\mathbf{w}_{ki}} R = \alpha_k \mathbf{w}_{ki}, \quad \frac{\partial R}{\partial \alpha_k} = \frac{1}{2} \sum_{i=1}^m \mathbf{w}_{ki}^\dagger \mathbf{w}_{ki} - \frac{m\tilde{m}}{2\alpha_k}, \quad \frac{\partial R}{\partial p_k} = \frac{\partial R}{\partial \beta_{ki}} = 0. \quad (\text{A.16})$$

The derivatives of $\ln \det \mathbf{H}$ will be derived in the next section: equations A.30, A.31, A.33, and A.34. Inserting this into equations 2.12 and 2.13 and applying equation A.6 leads to the regularized EM algorithm, equations 2.21 through 2.24 and equations 3.3 and 3.4.

¹⁶ The second term on the right of equation A.13 stems from the constraint $\sum_{k=1}^K p_k = 1 \Rightarrow p_K = 1 - \sum_{k=1}^{K-1} p_k$.

A.2 The Hessian. The derivations presented in this section are based on the following identity:

Lemma.

$$\text{If } \mathbf{q} = \mathbf{q}', \text{ then } \frac{\partial^2 S(\mathbf{q}|\mathbf{q}')}{\partial q_i \partial q_k} = - \left\langle \frac{\partial \Psi}{\partial q_i} \frac{\partial \Psi}{\partial q_k} \right\rangle_{\Lambda|\mathbf{D}, \mathbf{q}'} + \frac{\partial E_o}{\partial q_i} \frac{\partial E_o}{\partial q_k}. \quad (\text{A.17})$$

Proof. From the definition of S , equation A.4, we obtain

$$\begin{aligned} \frac{\partial^2 S(\mathbf{q}|\mathbf{q}')}{\partial q_i \partial q_k} &= \frac{\partial^2}{\partial q_i \partial q_k} \int (\ln P(\Lambda|\mathbf{D}, \mathbf{q})) P(\Lambda|\mathbf{D}, \mathbf{q}') d\Lambda \\ &= \int \frac{1}{P(\Lambda|\mathbf{D}, \mathbf{q})} \frac{\partial^2 P(\Lambda|\mathbf{D}, \mathbf{q})}{\partial q_i \partial q_k} P(\Lambda|\mathbf{D}, \mathbf{q}') d\Lambda \\ &\quad - \int \frac{1}{(P(\Lambda|\mathbf{D}, \mathbf{q}))^2} \frac{\partial P(\Lambda|\mathbf{D}, \mathbf{q})}{\partial q_i} \frac{\partial P(\Lambda|\mathbf{D}, \mathbf{q})}{\partial q_k} P(\Lambda|\mathbf{D}, \mathbf{q}') d\Lambda. \end{aligned}$$

For $\mathbf{q} = \mathbf{q}'$, the first integral in the last equation is zero and therefore

$$\frac{\partial^2 S(\mathbf{q}|\mathbf{q}')}{\partial q_i \partial q_k} = - \int \frac{\partial \ln P(\Lambda|\mathbf{D}, \mathbf{q})}{\partial q_i} \frac{\partial \ln P(\Lambda|\mathbf{D}, \mathbf{q})}{\partial q_k} P(\Lambda|\mathbf{D}, \mathbf{q}') d\Lambda.$$

Now use (i) $P(\Lambda|\mathbf{D}, \mathbf{q}) = \frac{P(\Lambda, \mathbf{D}|\mathbf{q})}{P(\mathbf{D}|\mathbf{q})} \Rightarrow \ln P(\Lambda|\mathbf{D}, \mathbf{q}) = \ln P(\Lambda, \mathbf{D}|\mathbf{q}) - \ln P(\mathbf{D}|\mathbf{q})$ and (ii) the fact that for $\mathbf{q} = \mathbf{q}'$, the following identity holds:

$$\begin{aligned} &\int \frac{\partial \ln P(\Lambda, \mathbf{D}|\mathbf{q})}{\partial q_i} P(\Lambda|\mathbf{D}, \mathbf{q}') d\Lambda \\ &= \int \frac{1}{P(\Lambda|\mathbf{D}, \mathbf{q}) P(\mathbf{D}|\mathbf{q})} \frac{\partial P(\Lambda, \mathbf{D}|\mathbf{q})}{\partial q_i} P(\Lambda|\mathbf{D}, \mathbf{q}') d\Lambda \\ &= \frac{1}{P(\mathbf{D}|\mathbf{q})} \frac{\partial}{\partial q_i} P(\mathbf{D}|\mathbf{q}) = \frac{\partial \ln P(\mathbf{D}|\mathbf{q})}{\partial q_i}. \end{aligned}$$

This gives

$$\begin{aligned} \frac{\partial^2 S(\mathbf{q}|\mathbf{q}')}{\partial q_i \partial q_k} &= - \int \frac{\partial \ln P(\Lambda, \mathbf{D}|\mathbf{q})}{\partial q_i} \frac{\partial \ln P(\Lambda, \mathbf{D}|\mathbf{q})}{\partial q_k} P(\Lambda|\mathbf{D}, \mathbf{q}') d\Lambda \\ &\quad + \frac{\partial \ln P(\mathbf{D}|\mathbf{q})}{\partial q_i} \frac{\partial \ln P(\mathbf{D}|\mathbf{q})}{\partial q_k}, \end{aligned}$$

which leads, with definitions 2.3 and A.2 to A.17.

It is now easy to prove the following relation, which allows the calculation of the Hessian of the total cost function E from U :

Theorem.

$$\text{If } \mathbf{q} = \mathbf{q}' \text{ then } \frac{\partial^2 E}{\partial q_i \partial q_k} = \frac{\partial^2 U}{\partial q_i \partial q_k} + \frac{\partial U}{\partial q_i} \frac{\partial U}{\partial q_k} - \left\langle \frac{\partial \Psi}{\partial q_i} \frac{\partial \Psi}{\partial q_k} \right\rangle_{\Lambda | \mathbf{D}, \mathbf{q}'} + \frac{\partial^2 R}{\partial q_i \partial q_k}. \quad (\text{A.18})$$

Proof. From equations 2.5 and A.5, we get:

$$\frac{\partial^2 E(\mathbf{q})}{\partial q_i \partial q_k} = \frac{\partial^2 U(\mathbf{q} | \mathbf{q}')}{\partial q_i \partial q_k} + \frac{\partial^2 S(\mathbf{q} | \mathbf{q}')}{\partial q_i \partial q_k} + \frac{\partial^2 R(\mathbf{q})}{\partial q_i \partial q_k}. \quad (\text{A.19})$$

Now making use of the above lemma, equation A.17, this leads to

$$\frac{\partial^2 E(\mathbf{q})}{\partial q_i \partial q_k} = \frac{\partial^2 U(\mathbf{q} | \mathbf{q}')}{\partial q_i \partial q_k} - \left\langle \frac{\partial \Psi}{\partial q_i} \frac{\partial \Psi}{\partial q_k} \right\rangle_{\Lambda | \mathbf{D}, \mathbf{q}'} + \frac{\partial E_o}{\partial q_i} \frac{\partial E_o}{\partial q_k} + \frac{\partial^2 R(\mathbf{q})}{\partial q_i \partial q_k}. \quad (\text{A.20})$$

Since $\mathbf{q} = \mathbf{q}'$, $\frac{\partial E_o}{\partial q_i}$ can be replaced by $\frac{\partial U(\mathbf{q} | \mathbf{q}')}{\partial q_i}$ due to equation A.6. This completes the proof.

From equations A.14 and A.16, we obtain:

$$\nabla_{\mathbf{w}_{ki}} \nabla_{\mathbf{w}_{k'j}}^\dagger U = \delta_{kk'} \delta_{ij} \beta_{ki} \mathbf{G} \mathbf{\Pi}_k \mathbf{G}^\dagger \quad (\text{A.21})$$

$$\nabla_{\mathbf{w}_{ki}} \nabla_{\mathbf{w}_{k'j}}^\dagger R = \delta_{kk'} \delta_{ij} \alpha_k \mathbf{I}. \quad (\text{A.22})$$

To calculate the outer product $\langle \nabla_{\mathbf{w}} \Psi (\nabla_{\mathbf{w}} \Psi)^\dagger \rangle$, take the gradient of the expression in equation A.9 and apply equation 1.1:

$$\nabla_{\mathbf{w}_{ki}} \Psi = - \sum_{t=1}^N \lambda_k(t) \beta_{ki} (y_{ti} - f(\mathbf{x}_t; \mathbf{w}_{ki})) \mathbf{g}(\mathbf{x}_t). \quad (\text{A.23})$$

In what follows, we will find expressions of the form $\langle [\sum_t \lambda_k(t) \Phi_{ki}(t)] [\sum_{t'} \lambda_{k'}(t') \Phi_{k'j}(t')] \rangle$. For independent training data, events at different times $t \neq t'$ are independent. Therefore $\langle \lambda_k(t) \lambda_{k'}(t') \rangle = \langle \lambda_k(t) \rangle \langle \lambda_{k'}(t') \rangle = \pi_k(t) \pi_{k'}(t')$. For $t = t'$, however, the events are not independent since at any particular time, only one of the indicator variables $\lambda_k(t)$ is “switched on.” Moreover, since the $\lambda_k(t)$ are binary, $[\lambda_k(t)]^2 = \lambda_k(t)$. This can be summarized as $\lambda_k(t) \lambda_{k'}(t) = \delta_{kk'} \lambda_k(t)$ ($\delta_{kk'}$ denotes the Kronecker delta), yielding the overall relation

$$\langle \lambda_k(t) \lambda_{k'}(t') \rangle = (1 - \delta_{tt'}) \pi_k(t) \pi_{k'}(t') + \delta_{tt'} \delta_{kk'} \pi_k(t). \quad (\text{A.24})$$

Using this result, we obtain

$$\begin{aligned}
& \left\langle \sum_t \lambda_k(t) \Phi_{ki}(t) \sum_{t'} \lambda_{k'}(t') \Phi_{k'i'}(t') \right\rangle \\
&= \sum_t \sum_{t'} \langle \lambda_k(t) \lambda_{k'}(t') \rangle \Phi_{ki}(t) \Phi_{k'i'}(t') \\
&= \sum_t \sum_{t'} \pi_k(t) \pi_{k'}(t') \Phi_{ki}(t) \Phi_{k'i'}(t') - \sum_t \pi_k(t) \pi_{k'}(t) \Phi_{ki}(t) \Phi_{k'i'}(t) \\
&\quad + \delta_{kk'} \sum_t \pi_k(t) \Phi_{ki}(t) \Phi_{k'i'}(t) \\
&= \left\langle \sum_t \lambda_k(t) \Phi_{ki}(t) \right\rangle \left\langle \sum_t \lambda_{k'}(t) \Phi_{k'i'}(t) \right\rangle - \sum_t \pi_k(t) \pi_{k'}(t) \Phi_{ki}(t) \Phi_{k'i'}(t) \\
&\quad + \delta_{kk'} \sum_t \pi_k(t) \Phi_{ki}(t) \Phi_{k'i'}(t),
\end{aligned}$$

and, after adding and subtracting a term $\delta_{kk'} \sum (\pi_k(t))^2 \Phi_{ki} \Phi_{k'i'}$,

$$\begin{aligned}
& \left\langle \sum_t \lambda_k(t) \Phi_{ki}(t) \sum_{t'} \lambda_{k'}(t') \Phi_{k'i'}(t') \right\rangle \\
&= \left\langle \sum_t \lambda_k(t) \Phi_{ki}(t) \right\rangle \left\langle \sum_t \lambda_{k'}(t) \Phi_{k'i'}(t) \right\rangle \\
&\quad - (1 - \delta_{kk'}) \sum_t \pi_k(t) \pi_{k'}(t) \Phi_{ki}(t) \Phi_{k'i'}(t) \\
&\quad + \delta_{kk'} \sum_t \pi_k(t) [1 - \pi_k(t)] \Phi_{ki}(t) \Phi_{k'i'}(t). \tag{A.25}
\end{aligned}$$

With equations A.23 and A.25, the definition $\Phi_{ki}(t) := \beta_{ki}(y_{ti} - f(\mathbf{x}_t; \mathbf{w}_{ki})) \mathbf{g}(\mathbf{x}_t)$, and the identity $\langle \Psi \rangle = U$ (from equation A.3), this leads to

$$\begin{aligned}
& \left\langle (\nabla_{\mathbf{w}_{ki}} \Psi) (\nabla_{\mathbf{w}_{k'i'}} \Psi)^\dagger \right\rangle \approx (\nabla_{\mathbf{w}_{ki}} U) (\nabla_{\mathbf{w}_{k'i'}} U)^\dagger \\
&\quad + \delta_{kk'} \delta_{i'i'} \beta_{ki}^2 \sum_{t=1}^N \pi_k(t) [1 - \pi_k(t)] (y_{ti} - f(\mathbf{x}_t; \mathbf{w}_{ki}))^2 \\
&\quad \times \mathbf{g}(\mathbf{x}_t) \mathbf{g}^\dagger(\mathbf{x}_t), \tag{A.26}
\end{aligned}$$

where the sums over cross-kernel terms, $k \neq k'$, and cross-coordinate terms, $i \neq i'$ have been neglected.¹⁷ Inserting equations A.21, A.22, and A.26 into

¹⁷ These terms can be both positive and negative and therefore tend to cancel each other out. Also note that for a distinct partitioning of the data among the kernels, the product $\pi_k(t) \pi_{k'}(t)$ for $k \neq k'$ will usually be small.

A.18 leads to the following expression for the Hessian $\mathbf{H} = [\nabla_{\mathbf{w}} \nabla_{\mathbf{w}}^\dagger E]_{\mathbf{w}=\hat{\mathbf{w}}}$:

$$\mathbf{H}_{kk'ii'} = \delta_{kk'} \delta_{ii'} (\mathbf{A}_{ki} + \alpha_k \mathbf{I}), \quad (\text{A.27})$$

where the definition of the \tilde{n} -by- \tilde{n} matrix,

$$\begin{aligned} \mathbf{A}_{ki} := & \beta_{ki} \mathbf{G} \mathbf{\Pi}_k \mathbf{G}^\dagger - \beta_{ki}^2 \sum_{t=1}^N \pi_k(t) [1 - \pi_k(t)] (y_{ti} - f(\mathbf{x}_t; \mathbf{w}_{ki}))^2 \\ & \times \mathbf{g}(\mathbf{x}_t) \mathbf{g}^\dagger(\mathbf{x}_t), \end{aligned} \quad (\text{A.28})$$

has been introduced. This matrix has a diagonal block structure, so for the logarithm of the determinant we obtain

$$\ln \det \mathbf{H} = \sum_{k=1}^K \sum_{i=1}^m \ln \det(\mathbf{A}_{ki} + \alpha_k \mathbf{I}), \quad (\text{A.29})$$

which gives the following derivatives:

$$\frac{\partial \ln \det \mathbf{H}}{\partial p_k} = 0 \quad (\text{A.30})$$

$$\begin{aligned} \frac{\partial \ln \det \mathbf{H}}{\partial \alpha_k} &= \frac{\partial}{\partial \alpha_k} \sum_{i=1}^m \ln \det(\mathbf{A}_{ki} + \alpha_k \mathbf{I}) = \sum_{i=1}^m \frac{\partial}{\partial \alpha_k} \sum_{v=1}^{\tilde{n}} \ln(\varepsilon_{ki}^v + \alpha_k) \\ &= \sum_{i=1}^m \sum_{v=1}^{\tilde{n}} \frac{1}{\varepsilon_{ki}^v + \alpha_k} = \frac{1}{\alpha_k} \sum_{i=1}^m \sum_{v=1}^{\tilde{n}} \frac{\alpha_k + \varepsilon_{ki}^v - \varepsilon_{ki}^v}{\varepsilon_{ki}^v + \alpha_k} \\ &= \frac{m\tilde{n} - \gamma_k}{\alpha_k} \end{aligned} \quad (\text{A.31})$$

$$\frac{\partial \ln \det \mathbf{H}}{\partial \beta_{ki}} = \frac{\partial}{\partial \beta_{ki}} \sum_{v=1}^{\tilde{n}} \ln(\varepsilon_{ki}^v + \alpha_k) = \sum_{v=1}^{\tilde{n}} \frac{1}{\varepsilon_{ki}^v + \alpha_k} \frac{\partial \varepsilon_{ki}^v}{\partial \beta_{ki}}, \quad (\text{A.32})$$

where ε_{ki}^v is the v th eigenvalue of \mathbf{A}_{ki} , and γ_k was defined in equation 2.20. The last expression can be simplified by assuming that the eigenvalues are homogeneous¹⁸ in β_{ki} , $\varepsilon_{ki}^v \propto \beta_{ki} \Rightarrow \frac{\partial \varepsilon_{ki}^v}{\partial \beta_{ki}} = \frac{\varepsilon_{ki}^v}{\beta_{ki}}$:

$$\frac{\partial \ln \det \mathbf{H}}{\partial \beta_{ki}} \approx \sum_{v=1}^{\tilde{n}} \frac{1}{\varepsilon_{ki}^v + \alpha_k} \frac{\varepsilon_{ki}^v}{\beta_{ki}} = \frac{\gamma_{ki}}{\beta_{ki}}. \quad (\text{A.33})$$

¹⁸ This is exact for $K = 1$, where the second contribution on the right of equation A.28 disappears. For $K > 1$, note that the terms in the sum on the right of equation A.28 scale like $\beta_{ki}^2 \pi_k(t) [1 - \pi_k(t)]$. For large kernel widths, $\beta_{ki} \ll 1$, they can obviously be neglected. For narrow kernels, the posteriors $\pi_k(t)$ will also be narrow, leading to a steep transition from $\pi_k(t) = 1$ to $\pi_k(t) = 0$. Because of the factor $\pi_k(t) [1 - \pi_k(t)]$, only exemplars with $\pi_k(t) \approx 0.5$ will give significant contributions, leaving the sum of order $\mathcal{O}(1)$, whereas the other term is of order $\mathcal{O}(N)$.

For MPPCA, where $\beta_{ki} = \beta_k \forall i$, we get

$$\frac{\partial \ln \det \mathbf{H}}{\partial \beta_k} = \frac{\partial}{\partial \beta_k} \sum_{i=1}^m \sum_{v=1}^{\tilde{n}} \ln(\varepsilon_{ki}^v + \alpha_k) \approx \sum_{i=1}^m \frac{\gamma_{ki}}{\beta_k} = \frac{\gamma_k}{\beta_k}. \quad (\text{A.34})$$

A.3 The Hessian for MPPCA. For MPPCA, we need to take the expectation value of the Hessian with respect to the distribution 3.9. In this study, the second term in the expansion of the Hessian, equation A.28, has been neglected, which is equivalent to the approximation made in Roberts et al. (1998). We then get

$$\begin{aligned} \mathbf{A}_{ki} &= \beta_k \langle \mathbf{G} \mathbf{\Pi}_k \mathbf{G}^\dagger \rangle \\ &= N_k \left[\beta_k \mathbf{M}_k^{-1} \mathbf{W}_k^\dagger \mathbf{S}_k \mathbf{W}_k + \mathbf{I} \right] \mathbf{M}_k^{-1}, \end{aligned} \quad (\text{A.35})$$

where in the second step, equation 3.23 has been applied.

A.4 A Special Case: Unconditional Densities. As mentioned in section 4.1, the modeling of unconditional densities is given by the special case $\mathbf{g}(\mathbf{x}_t) \equiv 1$. This allows a more accurate approximation to the Hessian, where cross-coordinate terms, $i \neq i'$, are not neglected, and equation A.27 is replaced by

$$H_{kk i i'} = \delta_{kk'} (A_{ki i'} + \delta_{i i'} \alpha_k) \quad (\text{A.36})$$

$$\begin{aligned} A_{ki i'} &:= \delta_{i i'} \beta_{ki} N_k \\ &\quad - \beta_{ki} \beta_{k i'} \sum_{t=1}^N \pi_k(t) [1 - \pi_k(t)] (y_{ti} - w_{ki})(y_{t i'} - w_{k i'}). \end{aligned} \quad (\text{A.37})$$

With the number of well-determined parameters, $\gamma_k = \sum_v \frac{\varepsilon_k^v}{\alpha_k + \varepsilon_k^v}$, where ε_k^v are the eigenvalues of \mathbf{A} in equation A.37, this leads to the following update rules (replacing equations 2.21–2.24):

$$\hat{p}_k = \frac{N_k}{N}, \quad \hat{w}_{ki} = \frac{\sum_t \pi_k(t) y_{ti} + (\alpha_k / \beta_{ki}) \mathcal{M}_i}{N_k + (\alpha_k / \beta_{ki})} \quad (\text{A.38})$$

$$\frac{1}{\hat{\beta}_{ki}} = \frac{\sum_t \pi_k(t) [y_{ti} - \hat{w}_{ki}]^2}{N_k - \gamma_k}, \quad \frac{1}{\hat{\alpha}_k} = \frac{1}{\gamma_k} \hat{\mathbf{W}}_k^\dagger \hat{\mathbf{W}}_k, \quad (\text{A.39})$$

where \mathcal{M} is the unconditional mean (this term stems from the modified prior mentioned in section 4.1), and N_k was defined in equation 2.25. Kernels with $N_k \leq \gamma_k$ are pruned.

Acknowledgments

Major parts of this work were carried out at King's College London and at Imperial College London, where I was supported by a Postgraduate Knight Studentship and by the Jefferiss Research Trust, respectively. I thank John G. Taylor, William D. Penny, and Stephen J. Roberts for stimulating discussions and helpful comments on earlier versions of the manuscript.

References

- Attias, H. (1999). Independent factor analysis. *Neural Computation*, 11(4), 803–851.
- Bishop, C. M. (1995). *Neural networks for pattern recognition*. New York: Oxford University Press.
- Bishop, C. M., & Qazaz, C. S. (1995). Bayesian inference of noise levels in regression. In *Proceedings ICANN 95* (pp. 59–64).
- Bishop, C. M., Svensen, M., & Williams, C. K. I. (1998). GTM: The generative topographic mapping. *Neural Computation*, 10(1), 215–234.
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B39(1), 1–38.
- Hoel, P. G. (1984). *Introduction to mathematical statistics*. New York: Wiley.
- Husmeier, D. (1998). *Modelling conditional probability densities with neural networks*. Unpublished doctoral dissertation, King's College London. Available online at: http://www.bioss.sari.ac.uk/~dirk/My_publications.html.
- Husmeier, D., & Taylor, J. G. (1997). Predicting conditional probability densities of stationary stochastic time series. *Neural Networks*, 10(3), 479–497.
- Husmeier, D., & Taylor, J. G. (1998). Neural networks for predicting conditional probability densities: Improved training scheme combining EM and RVFL. *Neural Networks*, 11(1), 89–116.
- Igelnik, B., & Pao, Y. H. (1995). Stochastic choice of basis functions on adaptive functional approximation and the functional-link net. *IEEE Transactions on Neural Networks*, 6, 1320–1329.
- Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3, 79–87.
- Jordan, M. I., & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6, 181–214.
- MacKay, D. J. C. (1992). Bayesian interpolation. *Neural Computation*, 4, 415–447.
- MacKay, D. J. C. (1993). Hyperparameters: Optimize, or integrate out. In G. Heidbreder (Ed.), *Maximum entropy and bayesian methods* (pp. 43–59). Norwell, MA: Kluwer.
- Ormoneit, D., & Tresp, V. (1996). Improved gaussian mixture density estimates using Bayesian penalty terms and network averaging. In D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (Eds.), *Advances in neural information processing systems*, 8 (pp. 542–548). Cambridge, MA: MIT Press.

- Papoulis, A. (1991). *Probability, random variables, and stochastic processes* (3rd ed.). New York: McGraw-Hill.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1992). *Numerical recipes in C*. New York: Cambridge University Press.
- Ripley, B. D. (1994). Neural networks and related methods for classification. *Journal of the Royal Statistical Society B*, 56(3), 409–456.
- Roberts, S. J., Husmeier, D., Rezek, I., & Penny, W. (1998). Bayesian approaches to gaussian mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Learning*, 20, 1133–1142.
- Spyers-Ashby, J. M. (1996). *The recording and analysis of tremor in neurological disorders*. Unpublished doctoral dissertation, Imperial College, London.
- Tipping, M. E., & Bishop, C. M. (1999). Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2), 443–482.
- Xu, L., & Jordan, M. I. (1996). On convergence properties for the EM algorithm. *Neural Computation*, 8, 129–151.

Received May 27, 1998; accepted November 12, 1999.

This article has been cited by:

1. Mark Girolami, Ben Calderhead. 2011. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **73**:2, 123-214. [[CrossRef](#)]
2. Antonio Penalver Benavent, Francisco Escolano Ruiz, Juan Manuel Saez. 2009. Learning Gaussian Mixture Models With Entropy-Based Criteria. *IEEE Transactions on Neural Networks* **20**:11, 1756-1771. [[CrossRef](#)]
3. L.P. Wang, C.R. Wan. 2008. Comments on “The Extreme Learning Machine”. *IEEE Transactions on Neural Networks* **19**:8, 1494-1495. [[CrossRef](#)]