Chapter 1

The Bayesian Paradigm: second generation neural computing

William D. Penny, Dirk Husmeier and Stephen J. Roberts[*]

When reasoning in the presence of uncertainty there is a unique and self-consistent set of rules for induction and model selection - Bayesian inference. Recent advances in neural networks have been fuelled by the adoption of this Bayesian framework, either implicitly, for example through the use of committees, or explicitly through Bayesian evidence and sampling frameworks. In this chapter, we show how this 'second generation' of neural network techniques can be applied to biomedical data and focus on the networks' ability to provide assessments of the confidence associated with its predictions. This is an essential requirement for any automatic biomedical pattern recognition system. It allows low confidence decisions to be highlighted and deferred, possibly to a human expert, and falls naturally out of the Bayesian framework.

1.1 Introduction

The Bayesian approach to learning in neural networks, proposed by Mackay[1] and Neal[2], has delivered a new conceptual framework that puts the study of neural nets on a sound theoretical footing. Together with the recent books by Bishop [3] and Ripley [4], which place neural networks in the context of statistical pattern recognition, these developments constitute what may be termed a second generation of neural computing. The practical benefits of the Bayesian

[*] Department of Electrical & Electronic Engineering, Imperial College, London SW7 2BT. Email s.j.roberts@ic.ac.uk

approach include principled methods for regularisation, feature selection, model selection, active learning and the calculation of error bars.

In contrast to the popular maximum likelihood framework, which aims to find a set of weights which minimise an error function, the Bayesian approach aims to integrate over all possible sets of weights. There are two main methods for doing this (i) the evidence method which performs the integration using an approximate analytic solution and (ii) the Hybrid Monte Carlo method which performs a numerical integration.

In this paper we have space to consider only the evidence framework. We show how it can be extended by considering committees of networks and focus on the handling of uncertainty. We look at a biomedical case study as an example.

1.2 Theory

Consider a data set $\mathbf{D} = \{(\mathbf{x}_t, y_t)\}_{t=1}^{N}$ generated by some unknown process, where $\mathbf{x}_t$ is an m-dimensional vector of explanatory variables, and y_t is a scalar[†] dependent variable or 'target'. In *regression* problems, y_t is continuous, and the interpolant is modelled by the network output $f(\mathbf{x};\mathbf{w})$, where $\mathbf{w}$ is a vector of network weights. If we assume that the targets are corrupted by additive Gaussian noise which is independent and identically distributed with variance $1/\beta$, then the probability of observing y_t conditional on the input vector $\mathbf{x}_t$ is

$$p(y_t|\mathbf{x}_t, \mathbf{w}) = \sqrt{\frac{\beta}{2\pi}} \exp\left(-\frac{\beta}{2}[y_t - f(\mathbf{x}_t;\mathbf{w})]^2\right) \tag{1.1}$$

Note that this expression depends on the network weights and a further so-called 'hyperparameter' β. In a two-class *classification* problem, the target variable y_t is binary (representing one of the two classes $\{\mathcal{C}_1, \mathcal{C}_2\}$), and the network output $f(\mathbf{x};\mathbf{w})$ represents the conditional probability for class $\mathcal{C}_1$,

$$p(y_t = 1|\mathbf{x}_t, \mathbf{w}) = f(\mathbf{x};\mathbf{w}). \tag{1.2}$$

The probability of y_t is

$$p(y_t|\mathbf{x}_t, \mathbf{w}) = f(\mathbf{x};\mathbf{w})^{y_t}[1 - f(\mathbf{x};\mathbf{w})]^{1-y_t}. \tag{1.3}$$

Note that, unlike the regression case, this expression does not depend on a further hyperparameter.

For regression problems we consider the use of Multilayer Perceptrons (MLPs) consisting of a layer or layers of sigmoidal or hyperbolic tangent nodes followed by a linear output layer. For classification problems the same network structure is used but the output layer generates the network 'activation' $a(\mathbf{x};\mathbf{w})$. The

[†] For simplicity of exposition we assume that we have only one network output although the theory is valid for multiple outputs.

final output network output, $f(\mathbf{x};\mathbf{w}) = g(a(\mathbf{x};\mathbf{w}))$ where g(a) is the sigmoid function

$$g(a) = \frac{1}{1+\exp(-a)} \tag{1.4}$$

For independent observations

$$P(\mathbf{D}|\mathbf{w}) = \prod_{t=1}^{N} p(y_t = 1|\mathbf{x}_t, \mathbf{w}) \tag{1.5}$$

A standard training scheme is to find the weights $\mathbf{w}_{ML}$ such that the likelihood $P(\mathbf{D}|\mathbf{w})$ is maximised. By defining the error function as the negative log-likelihood

$$E(\mathbf{w}) \;=\; -\ln P(\mathbf{D}|\mathbf{w}) \tag{1.6}$$

we see that

$$P(\mathbf{D}|\mathbf{w}) \;=\; \exp[-E(\mathbf{w})] \tag{1.7}$$

holds. The maximisation of the likelihood $P(\mathbf{D}|\mathbf{w})$ is equivalent to the minimisation of the error function E. The maximum likelihood prediction is

$$P(y|\mathbf{x},\mathbf{D}) \quad = \quad P(y|\mathbf{x},\hat{\mathbf{w}}_{ML}) \tag{1.8}$$

where $\hat{\mathbf{w}}_{ML}$ is the maximum likelihood, or minimum error, weight vector. This is found by a standard optimisation algorithm such as conjugate gradients. A disadvantage of the maximum likelihood estimator, however, is that the generalisation performance is poor when there is little training data.

1.2.1 Bayesian learning

A Bayesian analysis of network learning, however, shows that the best prediction that can be obtained on the basis of an observed training set $\mathbf{D}$ is the probability of y conditional on the input vector $\mathbf{x}$ and the training data $\mathbf{D}$. This is obtained by integrating over the network weights $\mathbf{w}$

$$P(y|\mathbf{x},\mathbf{D}) \quad = \quad \int P(y|\mathbf{x},\mathbf{w})P(\mathbf{w}|\mathbf{D})d\mathbf{w}. \tag{1.9}$$

where $P(\mathbf{w}|\mathbf{D})$ is the posterior distribution. This reflects our knowledge that a number of network solutions are consistent with the given training set. Comparison with equation 1.8 shows that, instead of making a prediction from a single network (the one with the lowest error), the Bayesian estimator combines predictions from many networks (an infinite number) where each prediction is weighted by the posterior probability.

Bayes rule says that the posterior probability is proportional to the likelihood of the model (how well it fits the data) and to the prior probabilitiy of the model i.e.

$$P(\mathbf{w}|\mathbf{D}) \propto P(\mathbf{D}|\mathbf{w})P(\mathbf{w}) \tag{1.10}$$

If we define a function $R(\mathbf{w})$ as the negative log of the prior probability and a function $C(\mathbf{w})$ as the negative log of the posterior probability then application of Bayes' rule leads to

$$C(\mathbf{w}) \quad = \quad E(\mathbf{w}) + R(\mathbf{w}) \tag{1.11}$$

The functions $C(\mathbf{w})$ and $R(\mathbf{w})$ can now be understood in terms of a more standard neural network approach. The function $R(\mathbf{w})$ is equivalent to a regularisation term, and $C(\mathbf{w})$ is equivalent to the total or 'regularised' error. The minimum regularised error network corresponds to the maximum posterior solution, $\hat{\boldsymbol{w}}_{MP}$. Moreover, if we define the prior distribution $P(\mathbf{w})$ as an isotropic Gaussian with variance $\frac{1}{\alpha}$ then

$$R(\mathbf{w}) \ = \frac{\alpha}{2}\sum_{i=1}^{W} w_i^2 \tag{1.12}$$

where W is the number of weights in the network and w_i is an individual weight. That is, the choice of a Gaussian prior corresponds to the use of a weight decay regulariser. Importantly, however, there are methods for estimating the weight decay coefficient, α, without resorting to cross-validation. These are described in the next two sections. Also, the use of different priors results in different regularizers. A more general prior, for example, is a product of isotropic Gaussians where the weights in the network are split up into different groups. This scheme can be applied such that the group of weights leaving each input has its own regularizer. The resulting method, called Automatic Relevance Determination (ARD), performs soft feature selection. As the focus of this paper is on error bars, however, we will consider the case of a single regularizer only.

1.2.2 The evidence framework

Because the prior distribution is dependent on α and, for regression problems, the likelihood is dependent upon β, a full Bayesian solution to the prediction problem is to augment equation 1.27 by also integrating over the posterior distribution $P(\alpha, \beta|\mathbf{D})$ as well as over the weights

$$P(y|\mathbf{x}, \mathbf{D}) \quad = \quad \int P(y|\mathbf{w}, \mathbf{x}) P(\mathbf{w}|\alpha, \beta, \mathbf{D}) P(\alpha, \beta|\mathbf{D}) d\mathbf{w} d\alpha d\beta \tag{1.13}$$

The evidence approach to Bayesian modelling, introduced to the neural network community by MacKay [1], seeks an analytic solution to the above equation by introducing two approximation steps. Firstly, the density $P(\alpha, \beta|\mathbf{D})$ is assumed to be unimodal and sharply peaked about its mode $\hat{\alpha}, \hat{\beta}$. This results in a collapse of the integral to

$$P(y|\mathbf{x}, \mathbf{D}) \quad = \quad \int P(y|\mathbf{w}, \mathbf{x}) P(\mathbf{w}|\hat{\alpha}, \hat{\beta}, \mathbf{D}) d\mathbf{w} \tag{1.14}$$

where $\hat{\alpha}$ and $\hat{\beta}$ are set to their maximum posterior values (more on this later). Secondly, the posterior density is approximated as $N(\hat{\mathbf{w}}_{MP}, \mathbf{A}^{-1})$ where $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicates a multivariate Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. The vector, $\hat{\mathbf{w}}_{MP}$ is the maximum posterior weight vector and $\mathbf{A}$ is the Hessian matrix

$$\mathbf{A} = [\nabla\nabla C]_{\hat{\mathbf{w}}_{MP}} \tag{1.15}$$

Error bars

For regression problems we can estimate error bars by making the further approximation that the network output can be written as a first order Taylor series expansion about $\hat{\mathbf{w}}_{MP}$. By substituting the posterior $N(\hat{\mathbf{w}}_{MP}, \mathbf{A}^{-1})$ into the integral 1.14 we can evaluate the output distribution analytically [1] as $P(y|\mathbf{x}, \mathbf{D}) = N(\bar{y}, \sigma^2)$ where $\bar{y} = f(\boldsymbol{x}; \hat{\mathbf{w}}_{MP})$ and

$$\sigma^2 = \frac{1}{\beta} + \boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{A}^{-1} \boldsymbol{g}(\boldsymbol{x}) \tag{1.16}$$

where $\boldsymbol{g}(\boldsymbol{x}) = [\partial f(\boldsymbol{x}; \boldsymbol{w})/\partial \boldsymbol{w}]_{\hat{\mathbf{w}}_{MP}}$. The error bars are given by the standard deviation of this output distribution, σ, and are seen to consist of two components. The first component is due to the intrinsic noise on the targets and the second component is due to uncertainty in the weight vector. The second component is input-dependent and, as we shall see in the results section, is larger for input patterns further away from the training set.

Moderated outputs

For classification problems a Taylor series expansion of the activation shows that the activation distribution is given by $P(a|\mathbf{x}, \mathbf{D}) = N(\bar{a}, s^2)$ where $\bar{a} = a(\boldsymbol{x}; \hat{\mathbf{w}}_{MP})$ and

$$s^2 = \tilde{\boldsymbol{g}}(\boldsymbol{x})^T \boldsymbol{A}^{-1} \tilde{\boldsymbol{g}}(\boldsymbol{x}) \tag{1.17}$$

where $\tilde{\boldsymbol{g}}(\boldsymbol{x}) = [\partial a(\boldsymbol{x}; \boldsymbol{w})/\partial \boldsymbol{w}]_{\hat{\mathbf{w}}_{MP}}$. The output distribution is related to the activation distribution by

$$P(y|\mathbf{x}, \mathbf{D}) = \int g(a) P(a|\mathbf{x}, \mathbf{D}) da \tag{1.18}$$

This integral cannot be evaluated analytically but is accurately approximated by [1]

$$P(y|\mathbf{x}, \mathbf{D}) = g(K(s)\bar{a}) \tag{1.19}$$

where

$$K(s) = \left(1 + \frac{\pi s^2}{8}\right)^{-1/2} \tag{1.20}$$

It is important to note that the above probability is not equal to $f(\mathbf{x}; \mathbf{w}_{MP})$. In fact, $P(y|\mathbf{x}, \mathbf{D})$ is nearer to 0.5 than is $f(\mathbf{x}; \hat{\mathbf{w}}_{MP})$ by an amount which is proportional to the posterior uncertainty on the network weights. To highlight this difference $P(y|\mathbf{x}, \mathbf{D})$ is refered to as the 'moderated output'.

Regularisation

In the evidence framework it is also assumed that the priors over α and β are constant. The maximum posterior estimate is therefore equivalent to the maximum likelihood estimate. This likelihood, however, refers to likelihood of the data after the weights have been integrated out. For example

$$P(\mathbf{D}|\alpha) = \int P(\mathbf{D}|\mathbf{w})P(\mathbf{w}|\alpha)d\mathbf{w} \tag{1.21}$$

This likelihood is also referred to as the evidence for α. Hence the name 'evidence framework'. The following formulae can be derived for these maximum evidence estimates, [1]

$$\hat{\alpha} = \frac{\gamma}{\sum_{i=1}^{W} w_i^2} \tag{1.22}$$

$$\hat{\beta} = \frac{N-\gamma}{\sum_{i=1}^{N}[y_i - f(\mathbf{x};\mathbf{w}_{MP})]^2} \tag{1.23}$$

where γ is given by

$$\gamma = \sum_{i=1}^{W} \frac{\lambda_i}{\lambda_i + \hat{\alpha}} \tag{1.24}$$

and λ_i are eigenvalues of $\boldsymbol{A}^{-1}$ and in equation 1.24 the old estimate for $\hat{\alpha}$ is used. The complete training/regularisation algorithm then consists of the following iterative scheme

- Given $\hat{\alpha}$ and $\hat{\beta}$ find the weight vector $\hat{\mathbf{w}}$ which minimises the total error function $C(\mathbf{w})$. This can be implemented with a standard optimisation algorithm such as conjugate gradients.

- Given $\hat{\mathbf{w}}$, re-estimate the hyperparameters $\hat{\alpha}$ and $\hat{\beta}$ according to equations 1.22 and 1.23.

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

1.2.3 Committees

The idea of integrating over weight space to obtain an optimal network prediction can be extended to integrating over different network models, m

$$P(y|\mathbf{x},\mathbf{D}) = \int P(y|\mathbf{x},\mathbf{D},m)P(m|\mathbf{D})dm \tag{1.25}$$

In practice this integral can be approximated by a committee

$$P(y|\mathbf{x},\mathbf{D}) \approx \sum_{i=1}^{M} P(y|\mathbf{x},\mathbf{D},m_i)P(m_i|\mathbf{D}) \tag{1.26}$$

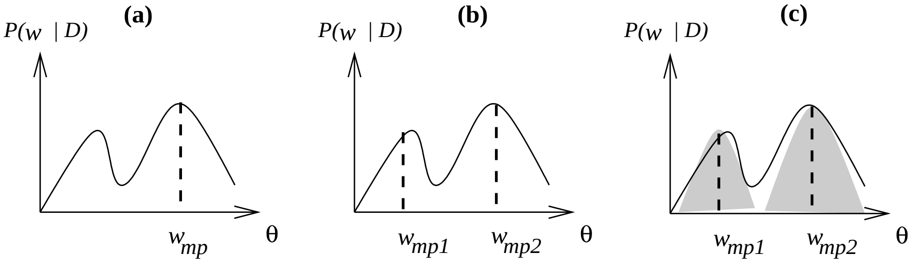

Figure 1.1: *The posterior distribution as estimated by (a) single model, (b) mutiple models and (c) local Gaussian approximations around each mode in a committee.*

The volume of the posterior distribution taken into account with this approximation is shown pictorially in Figure 1.1. For regression problems we can re-write the committee equation as

$$\hat{y} = \sum_{i=1}^{M} c_i y_i \tag{1.27}$$

where $c_i = P(m_i|\mathbf{D})$ and y_i is distributed as $P(y|\mathbf{x}, \mathbf{D}, m_i)$. If the networks in the committee have been trained with the evidence framework then y_i is distributed as $N(\bar{y}_i, \sigma_i^2)$ where $\bar{y}_i$ and σ_i^2 can be calculated from the error bars section in 1.2.2. The above equation is in the form of a mixture distribution. Although $\hat{y}$ is no longer a Gaussian we can still calculate its mean, y_{COM}, and variance σ_{COM}^2

$$y_{COM} = \sum_{i=1}^{M} c_i \bar{y}_i \tag{1.28}$$

$$\sigma_{COM}^2 = \sum_{i=1}^{M} c_i (\bar{y}_i - y_{COM})^2 + \sum_{i=1}^{M} c_i \sigma_i^2 \tag{1.29}$$

By substituting equation 1.16 for each committee member this can be re-written as

$$\sigma_{COM}^2 = \sum_{i=1}^{M} c_i (\bar{y}_i - y_{COM})^2 + \sum_{i=1}^{M} \frac{c_i}{\beta_i} + \sum_{i=1}^{M} c_i \boldsymbol{g}_i(\boldsymbol{x})^T \boldsymbol{A_i}^{-1} \boldsymbol{g}_i(\boldsymbol{x}) \tag{1.30}$$

which consists of three terms representing different contributions to the prediction error; (i) the disagreement among committee members, (ii) the target noise and (iii) the total weight uncertainty of the individual networks.

For classification problems the same analysis applies, but this time in the space of network activations. The resulting mean, a_{COM}, and variance, s^2_{COM}, are given by

$$a_{COM} = \sum_{i=1}^{M} c_i \bar{a}_i \tag{1.31}$$

$$s^2_{COM} = \sum_{i=1}^{M} c_i (\bar{a}_i - a_{COM})^2 + \sum_{i=1}^{M} c_i \tilde{\boldsymbol{g}}_i(\boldsymbol{x})^T \boldsymbol{A_i}^{-1} \tilde{\boldsymbol{g}}_i(\boldsymbol{x}) \tag{1.32}$$

The moderated committee output is then given by[‡]

$$y_{MOD} = g(K(s_{COM})a_{COM}) \tag{1.33}$$

For classification and regression networks trained by the evidence framework the mixing coefficients, c_i, can be obtained from estimates of the model evidence [5]. For small data sets, however, the estimates of model evidence are unreliable. For this reason c_i is often set to $1/M$ which is the approach adopted in this paper.

1.3 Example results

The Tremor data set, collected by Spyers-Ashby [6], is a two-class medical classification problem consisting of two input features derived from measurements of arm muscle tremor and a class label representing patient or non-patient. There are 178 training examples and 179 test examples. The patient population consisted of Parkinson's and multiple sclerosis patients and the non-patients were from a control group.

Figure 1.2 shows the training data set along with maximum posterior output (thise is given by the network output, y, or $1 - y$, whichever is the higher) from networks and committees of networks trained according to the Bayesian evidence framework. The unmoderated single network estimates a falsely high probability in regions of input space where it has seen little data (e.g. at the bottom of figure 1.2(a)). The moderated outputs in figure 1.2(b), however, are much less 'black and white' and correctly reflect our uncertainty in regions of low data density. This is demonstrated by computing the per-point negative log likelihood of the test data set, E_{test}. Of ten 3 hidden unit MLPs trained on this data the average value of E_{test} is 0.411 for unmoderated outputs, 0.407 for moderated outputs and for a committee of these same networks the moderated output gives a value of 0.375. Note that, in an example such as this, the use of

[‡] Strictly speaking this equation is no longer valid as the committee activations are drawn from a Gaussian Mixture not a Gaussian. However, a Gaussian which is moment-matched to the Gaussian Mixture (ie. same mean and variance) will give similar responses.

moderated outputs gives information regarding the uncertainty of the decisions. This is clearly a pre-requisite if any computerised method is to be used to aid patient diagnosis.

1.4 Summary

The Bayesian paradigm for learning in neural networks delivers principled methods for regularisation, feature selection, model selection, active learning and for the calculation of error bars and decision uncertainty. In this chapter we have had space to consider only the issue of uncertainty, however.

For the example classification problem the evidence framework provides a moderated output which gives a more conservative probability estimate in areas of low data density. Whilst this behaviour is qualitatively correct we also observe that the error bars and moderated outputs from committees of networks, rather than from a single network, are much more accurate. This is because the committee approach embodies a Gaussian Mixture approximation to the posterior distribution instead of a single Gaussian approximation.

Readers wishing to find out more about Bayesian methods for neural networks are referred, for feature selection to [2], for active learning to [1], for Hybrid Monte Carlo to [2] and for Bayesian methods applied to Radial Basis Functions to [7].

Bibliography

[1] D.J.C. Mackay. The evidence framework applied to classification networks. *Neural Computation*, 4(5):720–736, 1992.

[2] R. M. Neal. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer, New York, 1996.

[3] C. M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, Oxford, 1995.

[4] B. D. Ripley. Neural networks and related methods for classification. *Journal of the Royal Statistical Society B*, 56(3):409–456, 1994.

[5] D. Husmeier. *Modelling Conditional Probability Densities with Neural Networks*. PhD thesis, Department of Mathematics, King's College London, 1998.

[6] J.M. Spyers-Ashby P. Bain and S.J. Roberts. A comparison of fast Fourier transform and autoregressive spectral estimation techniques for the analysis of tremor data. *Journal of Neuroscience Methods*, 83:25–43, 1998.

[7] S. J. Roberts and W. D. Penny. A maximum certainty approach to feedforward neural networks. *Electronics Letters*, 33(4):306–307, 1998.

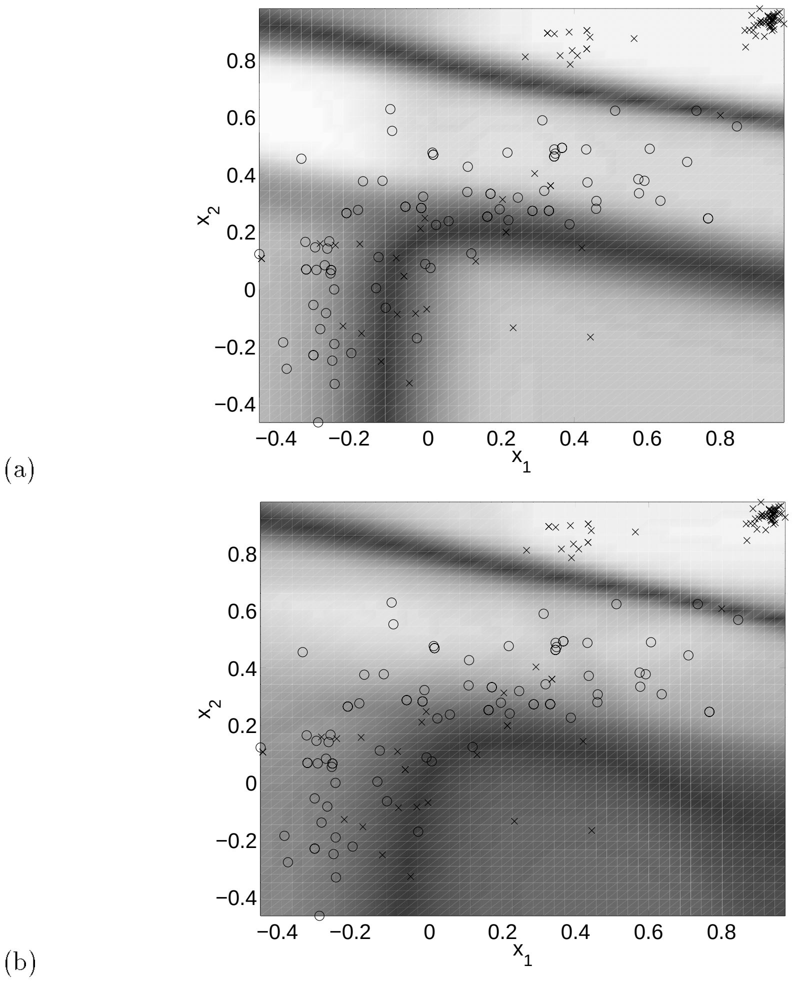

Figure 1.2: **Tremor data set**. *(a) Maximum posterior values from the unmoderated output of a 3 hidden-unit MLP, (b) Maximum posterior values from a moderated output of a committee of ten 3 hidden-unit MLPs. Crosses represent data points from patients, zeros indicate data points from normal subjects. The shade of grey codes the maximum posterior probability with dark grey being 0.5 and white being 1. Notice that the committee is less certain of its predictions in the bottom 'patient' region than is the single unmoderated network - for the committee this is a 'grey area' but the single unmoderated network is misleadingly confident.*