Helsinki University of Technology Laboratory of Computational Engineering Publications
Teknillisen korkeakoulun Laskennallisen tekniikan laboratorion julkaisuja
Espoo 2001 REPORT B29

BAYESIAN MODEL ASSESSMENT AND SELECTION USING EXPECTED UTILITIES

Aki Vehtari

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

Helsinki University of Technology Laboratory of Computational Engineering Publications
Teknillisen korkeakoulun Laskennallisen tekniikan laboratorion julkaisuja
Espoo 2001 REPORT B29

BAYESIAN MODEL ASSESSMENT AND SELECTION USING EXPECTED UTILITIES

Aki Vehtari

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Electrical and Communications Engineering, Helsinki University of Technology, for public examination and debate in Auditorium S3 at Helsinki University of Technology (Espoo, Finland) on the 14th of December, 2001, at 12 noon.

Helsinki University of Technology
Department of Electrical and Communications Engineering
Laboratory of Computational Engineering

Teknillinen korkeakoulu
Sähkö- ja tietoliikennetekniikan osasto
Laskennallisen tekniikan laboratorio

Distribution:
Helsinki University of Technology
Laboratory of Computational Engineering
P. O. Box 9400
FIN-02015 HUT
FINLAND
Tel. +358-9-451 4826
Fax. +358-9-451 4830
http://www.lce.hut.fi

Online in PDF format: http://lib.hut.fi/Diss/2001/isbn9512257653/

E-mail: Aki.Vehtari@hut.fi

ISBN 951-22-5764-5 (printed)
ISBN 951-22-5765-3 (PDF)
ISSN 1455-0474
PicaSet Oy
Espoo 2001

Abstract

In this work, we discuss practical methods for the Bayesian model assessment and selection based on expected utilities, and propose several new methods and techniques for the analysis of the models.

The Bayesian approach offers a consistent way to use probabilities to quantify uncertainty in inference resulting in a probability distribution expressing our beliefs regarding how likely the different predictions are. The use of Bayesian models in increasingly complex problems is facilitated by advances in Markov chain Monte Carlo methods and computing power.

A natural way to assess the goodness of the model is to estimate its future predictive capability by estimating expected utilities. With application specific utilities, the expected benefit or the cost of using the model can be readily computed. We propose an approach using cross-validation predictive densities to compute the expected utilities and Bayesian bootstrap to obtain samples from their distributions. Instead of just making a point estimate, it is important to estimate the distribution of the expected utility, as it describes the uncertainty in the estimate. The distributions of the expected utilities can also be used to compare models, for example, by computing the probability of one model having a better expected utility than some other model. The expected utilities take into account how the model predictions are going to be used and thus may reveal that even the best model selected may be inadequate or not practically better than the previously used models.

To make the model easier to analyse, or to reduce the cost of making measurements or computation, it may be useful to select a smaller set of input variables. Computing the cross-validation predictive densities for all possible input combinations easily becomes computationally prohibitive. We propose to use a variable dimension Markov chain Monte Carlo method to find out potentially useful input combinations, for which the final model choice and assessment is done using the cross-validation predictive densities.

We demonstrate the usefulness of the presented approaches with MLP neural networks and Gaussian process models in three challenging real-world case problems.

Preface

This thesis for the degree of Doctor of Technology has been prepared in the Laboratory of Computational Engineering at the Helsinki University of Technology during the years 1997-2001.

I want to express my sincere gratitude to Prof. Jouko Lampinen who has been excellent supervisor, and to Academy Prof. Kimmo Kaski who has provided first-rate computational resources. I also wish to thank Prof. Jari Kaipio and Prof. Henry Tirri for reviewing the thesis and for their comments and suggestions on the text.

I am grateful for the application specific expert knowledge provided by Dr. Hanna Järvenpää in the concrete quality prediction problem, by Dr. Jukka Heikkonen in the forest scene classification problem, and by Mr. Kimmo Leinonen in the electrical impedance tomography problem.

Many thanks are due to M.Sc. Simo Särkkä, who helped programming Matlab tools for handling FBM-models and MCMC convergence diagnostics, and Mr. Toni Tamminen, Mr. Ilkka Kalliomäki, M.Sc. Markus Varsta, M.Sc. Timo Kostiainen, and M.Sc. Jani Lahtinen, who have helped me proof-reading the text. I would also like to thank all the people in the laboratory for a very nice and inspiring atmosphere.

My studies were partly funded by Graduate School in Electronics, Telecommunications and Automation (GETA), the National Technology Agency (TEKES) Grant 40888/97 (project title: *PROMISE, Applications of Probabilistic Modeling and Search*) and the Academy of Finland under the Finnish Centre of Excellence Programme during 2000-2005 (project title: *Research Centre for Computational Science and Engineering*, project number 44897).

Aki Vehtari

Contents

Abstract **i**

Preface **iii**

Contents **v**

1 Introduction **1**

2 The Bayesian approach **5**

2.1 Introduction 5

2.1.1 Bayes' rule 5

2.1.2 Prediction 6

2.1.3 The role of prior knowledge in statistical models 7

2.1.4 Approximations of marginal distributions 10

2.1.5 Markov chain Monte Carlo 11

2.2 Models and priors 13

2.2.1 Residual models 13

2.2.2 MLP neural networks 16

2.2.3 Gaussian processes 18

2.2.4 The automatic relevance determination prior 19

2.2.5 On sensitivity to the prior distributions 23

2.3 Illustrative examples 25

2.3.1 Case I: Regression task in concrete quality estimation . . . 25

2.3.2 Case II: Classification task in forest scene analysis 27

2.3.3 Case III: Inverse problem in electrical impedance tomography 29

3 Model assessment and selection **33**

3.1 Introduction 33

3.1.1 Expected utilities 34

3.1.2 Cross-validation predictive densities 35

3.1.3 On assumptions made on future data distribution 36

3.2 Methods 38
3.2.1 Importance sampling leave-one-out cross-validation . . . 38
3.2.2 Importance weights for MLP and GP models 40
3.2.3 *k*-fold cross-validation 41
3.2.4 Estimating the distribution of the expected utility 42
3.2.5 Model comparison with expected utilities 44
3.3 Relations to other approaches 46
3.3.1 Prior predictive densities and Bayes factors 46
3.3.2 Posterior predictive densities 48
3.3.3 Other predictive densities 49
3.3.4 Information criteria and the effective number of parameters 49
3.4 Illustrative examples 55
3.4.1 Toy problem: MacKay's robot arm 55
3.4.2 Case I: Concrete quality estimation 63
3.4.3 Case II: Forest scene classification 69

4 Input variable selection 73
4.1 Introduction 73
4.2 Methods 75
4.2.1 Priors for input selection 75
4.2.2 Reversible jump Markov chain Monte Carlo 78
4.3 Illustrative examples 79
4.3.1 Case I: Concrete quality estimation 79
4.3.2 Case II: Forest scene classification 92

5 Conclusions 97

References 99

Chapter 1

Introduction

In this work, we discuss practical methods for Bayesian model assessment, comparison, and selection based on expected utilities, and propose several new methods and techniques for the analysis of the models.

The generalization capability of a statistical model, classical or Bayesian, is ultimately based on the prior assumptions. The Bayesian approach offers a consistent way to use probabilities to quantify uncertainty in inferences and the result of Bayesian inference is a probability distribution expressing our beliefs regarding how likely the different predictions are. In complex problems, one of the major advantages of the Bayesian approach is that we are not forced to guess unknown attributes, such as the number of degrees of freedom in the model, degree of nonlinearity of the model with respect to each input variable, or the exact form of the distribution of the model residuals. The Bayesian approach permits propagation of uncertainty in quantities that are unknown to other assumptions in the model, which may be more generally valid or easier to guess in the problem. Although no guesses are required for the exact values of the parameters or any smoothness coefficients or other hyperparameters, guesses are made for the exact forms of their distributions. The goodness of the model depends on these guesses, which in practical applications makes it necessary to carefully validate the models and assess their performance.

In Chapter 2, we give a short review of the Bayesian approach and the models and the priors we have used. We emphasise that it is impossible logically to distinguish between model assumptions and the prior distribution of parameters. The model *is* the prior in the wide sense that it is a probability statement of all the assumptions currently to be tentatively entertained *a priori*. All generalization is based on prior knowledge, that is, training samples provide information only at those points and the prior knowledge provides the necessary link between the training samples and the not yet measured future samples.

We also describe briefly the Markov chain Monte Carlo (MCMC) methods we have used to numerically approximate the required integrals in the Bayesian approach. We describe the MLP network and Gaussian process models, which are flexible nonlinear models used in our illustrative examples, and useful priors for them. We illustrate some specific issues of residual model selection and the elaborate input variable prior (“automatic relevance determination”) in MLP networks and Gaussian processes.

We demonstrate the benefits of the Bayesian approach and the models and priors reviewed in three challenging real world cases:

Case I: a regression problem of predicting the quality properties of concrete
Case II: a classification problem of recognizing tree trunks in forest scenes
Case III: an inverse-problem in electrical impedance tomography.

A common factor in the problems is that they have many potentially useful measurements with unknown relevance, noise components, correlations, nonlinear dependencies, and cross-effects. The first two cases are used also in later chapters to illustrate the discussion.

In Chapter 3, we discuss Bayesian model assessment, comparison, and selection using expected utilities. Whatever way the model building and the selection have been done, the goodness of the final model should be assessed in order to find out whether it is useful in a given problem. Even the best model selected from some collection of models may be inadequate or not considerably better than the previously used models. In practical problems, it is important to be able to describe the quality of the model in terms of the application field instead of statistical jargon. It is also important to be able to describe the uncertainty in our estimates.

In prediction problems, it is natural to assess the predictive ability of the model by estimating the expected utilities, that is, the relative values of consequences. Expected utilities describe how good predictions the model makes for future observations from the same process that generated the given set of training data. By using application specific utility functions, the expected benefit or cost of using the model for predictions (e.g., by financial criteria) can be readily computed. In lack of application specific utilities, many general discrepancy and likelihood utilities can be used. Expected utilities are also useful in non-prediction problems where the goal is just to get scientific insights in modeled phenomena. Maximizing the predictive likelihood utility for such model is same as minimizing information theoretic Kullback-Leibler divergence between the model and the unknown distribution of the data. The reliability of the estimated expected utility can be assessed by estimating its distribution.

We give a unified and formal presentation of how to estimate the distributions of the expected utilities from the Bayesian viewpoint. As the future observations are not yet available, we have to approximate the expected utilities by reusing samples we already have. We discuss the cross-validation predictive density approach, which can be used to compute (nearly unbiased) estimates of the expected utilities. We clearly state the assumptions made in the approach. For simple models, the cross-validation densities may be computed quickly using analytical solutions, but for more complex models we need to use approximations. We discuss the properties of two practical methods, importance sampling leave-one-out (IS-LOO-CV) and k-fold cross-validation (k-fold-CV). We discuss how the reliability of importance sampling can be estimated and if there is reason to suspect the reliability of the importance sampling, we suggest to use predictive densities from the k-fold-CV. We also note that the k-fold-CV has to be used if data points have certain dependencies. Since the k-fold-CV predictive densities are based on smaller data sets than the full data set, the expected utility estimate is biased. This bias has been usually ignored, but in the case of different steepness of the learning curves and in the model assessment, this bias should not be ignored. To get more accurate results we use a less well-known first order bias correction.

To assess the reliability of the estimated expected utility it is important to estimate its distribution. We discuss simple Gaussian approximations and propose a quick and generic approach based on the Bayesian bootstrap method, which makes a simple non-parametric approximation, for obtaining samples from the distributions of the expected utilities. The proposed approach can handle the variability due to Monte Carlo integration, bias correction estimation, and approximation of the future data distribution. Moreover, it works better than Gaussian approximation in the case of arbitrary summary quantities and non-Gaussian distributions.

If there is a collection of models under consideration, the distributions of the expected utilities can also be used for model comparison. With the proposed method, it is for example easy to compute the probability of one model having a better expected utility than some other model. Following simplicity postulate (aka parsimony principle), it is useful to start from simpler models and then test if more complex model would give significantly better predictions. An extra advantage of comparing the expected utilities is that even if there is high probability that one model is better, it might be found out that the difference between the expected utilities still is practically negligible. For example, it is possible that using statistically better model would save negligible amount of money.

We discuss the assumptions and restrictions in the approach and relations to approaches for comparison of methods. We note the relation of the expected predictive densities to Bayes factors and discuss relations to other predictive densities, of which most interesting ones are the prior and posterior predictive densities. We discuss relations to information criteria (e.g., AIC, NIC, DIC), which can also be used to estimate the expected utility of the model. We also discuss the concept of the effective number of parameters and describe how it can be estimated using the cross-validation approach.

In Chapter 4, we discuss problems specific in input selection. With suitable priors, it is possible to have a large number of input variables in Bayesian models, as less relevant inputs can have a smaller effect in the model. To make the model easier to analyse (easier to gain scientific insights), or to reduce the cost of making measurements or computation, it may be useful to select a smaller set of input variables. In addition, if the assumptions of the model and prior do not match well the properties of the data, reducing the number of input variables may even improve the performance of the model. Our goal is to find a model with the smallest number of input variables with statistically or practically at least the same expected utility as the full model with all the available inputs. In the case of input selection and K inputs, there are 2^K input combinations and computing the cross-validation predictive densities for each model easily becomes computationally prohibitive.

We propose to use the variable dimension Markov chain Monte Carlo methods to find out potentially useful input combinations, for which the final model choice and assessment is done using the cross-validation predictive densities. We discuss reversible jump Markov chain Monte Carlo (RJMCMC) method, which is one of the simplest and fastest variable dimension MCMC methods. The approach is based on the fact that the posterior probabilities of the model, given by the RJMCMC, are proportional to the product of the prior probabilities of the models and the prior predictive likelihoods of the models that can be used to estimate the lower limit of the expected cross-validation predictive likelihood. We also discuss different ways of including information about prior probabilities of the number of input variables. Additionally, in the case of very many inputs, we propose that instead of using the probabilities of input combinations, the marginal probabilities of inputs can be used to select potentially useful models. In addition to input selection, the marginal probabilities of inputs, given by the RJMCMC, can be used to estimate the relevance of the inputs, which has great importance in analyzing the final model.

Finally, in the last chapter a brief conclusion is drawn.

Chapter 2

The Bayesian approach

2.1 Introduction

In Bayesian data analysis, all uncertain quantities are modeled as probability distributions, and inference is performed by constructing the posterior conditional probabilities for the unobserved variables of interest, given the observed data and prior assumptions. Excellent references for Bayesian data analysis are, for example, (in increasing order of deepness) (Sivia, 1996; Gelman et al., 1995; Bernardo and Smith, 1994). For additional discussion about the concept of probability, see (Bayes, 1763; Laplace, 1825; Cox, 1946; Jeffreys, 1961; Jaynes, 1996).

We first give a short overview of the Bayesian approach and the Markov chain Monte Carlo methods used for the integrations. We then describe the residual models, the MLP networks, and Gaussian processes used in our examples, discussing some important issues specific to prior distributions.

We have tried to follow the notation of Gelman et al. (1995). For example, we use the terms *distribution* and *density* interchangeably, the same notation is used for continuous density functions and discrete probability mass functions, and we use the notation $r \sim F(a)$ as shorthand for $p(r) = F(r|a)$ where a denotes the parameters of the distribution F, and the random variable argument r is not shown explicitly.

2.1.1 Bayes' rule

The key principle of Bayesian approach is to construct the posterior probability distributions for all the unknown entities in a model, given the data. To use the model, marginal distributions are constructed for all those entities that we are interested in, that is, the end variables of the study. These can be the parameters in parametric models, or the predictions in (non-parametric) regression or classification tasks.

The use of the posterior distributions requires an explicit definition of the prior distributions for the parameters. The posterior probability distribution for the parameters θ in a model M given the data D is, according to the Bayes' rule,

$$p(\theta|D, M) = \frac{p(D|\theta, M)p(\theta|M)}{p(D|M)}, \tag{2.1}$$

where $p(D|\theta, M)$ is the likelihood of the parameters θ, $p(\theta|M)$ is the prior probability distribution of θ, and $p(D|M)$ is a normalizing constant, or the evidence of the model M. The term M denotes all the hypotheses and assumptions that are made in defining the model, such as a choice of MLP network, specific noise model etc. All the results are conditioned on these assumptions, and to make this clear we prefer to have the term M explicitly in the equations. In this notation the normalization term $p(D|M)$ is directly understandable as the marginal probability of the data, conditional on M,

$$p(D|M) = \int p(D|\theta, M)p(\theta|M)d\theta. \tag{2.2}$$

When having several models, $p(D|M_i)$ is the likelihood of the model M_i, which can be used to compare the probabilities of the models, hence the term evidence of the model. A widely used Bayesian model choice method between two models is based on Bayes Factors, $p(D|M_1)/p(D|M_2)$ (see section 3.3.1). The more common notation of the Bayes formula, with M dropped, causes more easily misinterpreting the denominator $p(D)$ as some kind of probability of obtaining the data D in the studied problem (or prior probability of the data before the modeling).

2.1.2 Prediction

The posterior predictive distribution of output y for the new input $x^{(n+1)}$ given the training data, $D = \{(x^{(i)}, y^{(i)}); i = 1, 2, \ldots, n\}$, is obtained by integrating the predictions of the model with respect to the posterior distribution of the model,

$$p(y|x^{(n+1)}, D, M) = \int p(y|x^{(n+1)}, \theta, D, M)p(\theta|D, M)d\theta, \tag{2.3}$$

where θ denotes all the model parameters and hyperparameters of the prior structures and M is all prior knowledge in the model specification (including all implicit and explicit prior specifications). If the predictions are independent of the training data given the parameters of the model (e.g., in parametric models) then $p(y|x^{(n+1)}, \theta, D, M) = p(y|x^{(n+1)}, \theta, M)$.

The predictive distributions can be used to make guesses. For example, with squared error loss the best guess for model prediction (with an additive zero-mean noise model) corresponds to the expectation of the posterior predictive distribution

$$\hat{y} = E_y[y|x^{(n+1)}, D, M]. \tag{2.4}$$

The predictive distribution also contains information about the uncertainty in the prediction, which is usually summarized by the variance or the quantiles of the predictive distribution of the output or the guess.

If the integral in Equation 2.3 is analytically intractable (e.g., in the examples in section 3.4), it can be approximated with the methods discussed in section 2.1.4.

2.1.3 The role of prior knowledge in statistical models

Describing the prior information explicitly distinguishes the Bayesian approach from other methods. It is important to notice, however, that the role of prior knowledge is equally important in any other approach, including the maximum likelihood method. It is impossible logically to distinguish between model assumptions and the prior distribution of parameters. The model *is* the prior in the wide sense that it is a probability statement of all the assumptions currently to be tentatively entertained *a priori* (Box, 1980). All generalization is based on prior knowledge, that is, training samples provide information only at those points and the prior knowledge provides the necessary link between the training samples and the not yet measured future samples (Lemm, 1996, 1999).

Recently, some important no-free-lunch (NFL) theorems have been proven, that help to understand this issue. Wolpert (1996a,b) showed that if the class of approximating functions is not limited, any learning algorithm (that is, a procedure for choosing the approximating function) can as readily perform worse or better than a random one, measured by off-training set (OTS) error, and averaged over loss functions. This theorem implies that it is not possible to find a learning algorithm that is *universally* better than a random one. In other words, if we do not assume anything *a priori*, the learning algorithm cannot learn anything from the training data that would generalize to the off-training set samples.

Wolpert and Macready (1995) and Wolpert (1996b) analysed the cross-validation method for model selection in more depth and showed that the NFL theorem also applies to cross-validation. The basic result in the papers is that without priors on functions, choosing the function by cross-validation performs on average as well as a random algorithm, or anti-cross-validation, where the function with the largest cross-validation error is selected. In practice this means that if cross-validation is used to choose from a very large (actually infinite) set of models, there is no guarantee of any generalization at all. This is easy to understand intuitively, as in such a situation the chosen algorithm is the one that happens to minimize the error on the whole training set, and if the set of algorithms is large there is a high chance that a well fitting (“over-fitted”) solution exists in the set. It should be noted, however, that due to computational limitations, cross-validation can in practice be used to choose between rather few models (typically less than thousands), so that the choice of the models imposes a very strict prior on the functions. Thus, the NFL theorems do not invalidate the use of cross-validation in practical model selection. The implications are more in principal, emphasizing that the *a priori* selection of plausible solutions is necessary when using cross-validation for model selection, and in this respect, cross-validation does not provide an alternative that would not require using prior knowledge in the modeling.

In practice, statistical models like parametric models or MLP networks probably contain more likely too strict priors rather than too little prior knowledge. For example, every discrete choice in the model, such as the Gaussian noise model, represents an infinite amount of prior information (Lemm, 1996). Any finite amount of information would not correspond to a probability of one for, for example, the Gaussian noise model and probability zero for all the other alternatives. In addition, the functional form of the model may be predetermined (as in polynomial fitting), or the number of degrees of freedom may be fixed (as in MLP networks trained with error minimization without regularization). Thus, there is also a large amount of prior information in maximum likelihood models, even though the model parameters are determined solely by the data, to maximize the likelihood $p(D|\theta, M)$, or to minimize the negative log-likelihood error function. Actually, the goodness of this prior knowledge is what separates "good" and "bad" maximum likelihood models.

In the Bayesian approach, a certain part of the prior knowledge is specified more explicitly, in the form of prior distributions for the model parameters, and hyperpriors for the parameters of the prior distributions. In complex models like MLP networks, the relation between the actual domain knowledge of the experts and the priors for the model parameters is not simple, and thus it may be in practice difficult to incorporate very sophisticated background information into the models via the priors of the parameters.

However, a considerable advantage of the Bayesian approach is that it gives a principled way to do inference when some of the prior knowledge is lacking or vague, and thus one is not forced to guess values for the attributes that are unknown. This is done by marginalization, or integrating over the posterior distribution of the unknown variables, as explained in the next section.

A lot of work has been done to find "non-informative" priors that could be used to specify a complete lack of knowledge of a parameter value. Some approaches are uniform priors, Jeffreys' prior (Jeffreys, 1961), and reference priors (Berger and Bernardo, 1992). See (Kass and Wasserman, 1996) for a review and (Yang and Berger, 1997) for a large catalog of different "non-informative" priors for various statistical models.

Among Bayesians, the use of "non-informative" priors is often referred as the "objective Bayesian approach", in contrast to the informative (subjective) priors, which reflect the subjective opinions of the model builder. However, in the light of the NFL theorems, this requires that the hypothesis space is already so constrained that it contains the sufficient amount of prior information that is needed to be able to learn a generalizing model (Lemm, 1999, ch. 2.7).

By using “non-informative” priors, the fixed, or guessed, choices can be moved to higher levels of hierarchical models. Goel and Degroot (1981) showed that in hierarchical models the training data contains less information of hyperparameters that are higher in the hierarchy, so that the prior and posterior for the hyperparameters tend to be more similar. Thus, the models are less sensitive to the choices made in higher levels, implying that higher level priors are in general less informative, and thus less subjective.

In this way, the hierarchical prior structure can be used to specify a partial lack of knowledge in a controllable way. For example, if it is difficult to choose between a Gaussian and a longer tailed (leptokurtic) noise model, one can include them both in the prediction model by using non-zero prior probabilities for the two noise models. The posterior probabilities of the noise models will be determined “objectively” from the match of the noise distribution and the realized model residuals. In section 2.2.1 we present an example of using Student’s t-distribution with an unknown number of degrees of freedom ν as the noise model (thus comprising near Gaussian and longer tailed distributions), and integrating over the posterior distribution of ν in predictions. Some advice on the design of the hierarchical prior structures and robust noise models can be found, for example, in (Gelman et al., 1995).

A typical attribute that is difficult to guess in advance in complex statistical models is the correct number of degrees of freedom, as it depends on the number of the training samples and their mutual correlation, distribution of the noise in the samples and the complexity of the underlying phenomenon to be modeled. In general, the complexity of the model cannot be defined by only one number, the total number of degrees of freedom, but instead the models have multiple dimension of complexity. In the Bayesian approach, one can use a vague prior for the total complexity (called the effective number of parameters), and use a hierarchical prior structure to allow different complexity in different parts of the model. For example, the parameters may be assigned to different groups, so that in each group the parameters are assumed to have the same hyperparameter, while different groups can have different hyperparameters. Then a hyperprior is defined to explain the distribution of all the hyperparameters. An example of this type of prior, called the Automatic Relevance Determination prior, is discussed in section 2.2.4.

Although in full hierarchical Bayesian models no guesses are made for exact values of the parameters or any smoothness coefficients or other hyperparameters, guesses have to be made for the exact forms of their distributions. The goodness of the model depends on the guesses that are usually based on uncertain assumptions, which in practical applications make it necessary to carefully validate the models, using, for example, Bayesian posterior analysis (Gelman et al., 1995, ch. 6), or the cross-validation approach discussed in Chapter 3. This also implies that in practice the Bayesian approach may be more sensitive to prior assumptions than the more classical methods. This is discussed in more detail in section 2.2.5.

2.1.4 Approximations of marginal distributions

The marginalization in Equation 2.3 often leads to complex integrals that cannot be solved in closed form, and thus there is a multitude of approaches that differ in how the integrals are approximated.

Closest to the conventional maximum likelihood approach is the maximum a posteriori (MAP) approach, where the posterior distribution of the parameters is not considered, but the parameters are sought to maximize the posterior probability $p(w|D, M) \propto p(D|w, M)p(w|M)$, or to minimize the negative log-posterior cost function

$$E = -\log p(D|w, M) - \log p(w|M). \tag{2.5}$$

The weight decay regularization in MLP networks is an example of this technique: for a Gaussian prior on the weights w the negative log-prior is $\gamma \sum_i w_i^2$. The main drawback of this approach is that it gives no tools for setting the hyperparameters due to lack of marginalization over these “nuisance parameters”. In the weight decay example, the variance term $1/\gamma$ must be guessed, or set with some external procedure, such as cross-validation.

In the empirical Bayesian approach, specific values are estimated for the hyperparameters. For MLP networks, this approach was introduced by MacKay (1992) in the evidence framework (also called type II maximum likelihood approach (Berger, 1985)). In this framework, the hyperparameters α are set to values that maximize the evidence of the model $p(D|\alpha, M)$, that is, the marginal probability of the data given the hyperparameters, integrated over the parameters, $p(D|\alpha, M) = \int p(D|w, M)p(w|\alpha, M)dw$. A Gaussian approximation is used for the posterior of the parameters $p(w|D, M)$, to facilitate closed form integration, and thus the resulting posterior of w is specified by the mean and variance of the Gaussian approximation.

In the full Bayesian approach, no fixed values are estimated for any parameters or hyperparameters. Approximations are then needed for the integrations over the hyperparameters to obtain the posterior of the parameters and over the parameters to obtain the predictions of the model, as shown in Equation 2.3. The correctness of the inference depends on the accuracy of the integration method, hence it depends on the problem which approximation method is appropriate. Methods for approximating the integrations in complex models include, for example, ensemble learning (Barber and Bishop, 1998), which aims to approximate the posterior distribution by minimizing the Kullback-Leibler divergence between the true posterior and a parametric approximating distribution, variational approximations (Jordan et al., 1998) for approximating the integration by a tractable problem, mean field approach (Winther, 1998), in which the problem is simplified by neglecting certain dependencies between the random variables, and the Markov Chain Monte Carlo techniques for numerical integration, discussed in more detail in the next section.

2.1.5 Markov chain Monte Carlo

If the marginalization in Equation 2.3 is analytically intractable (e.g., in examples in section 2.3), the expectation (or other summary quantity) of any function g can be estimated by using the Monte Carlo approximation

$$E_y[g(y)|x^{(n+1)}, D, M] \approx E_j[g(\dot{y}_j)] = \frac{1}{m}\sum_{j=1}^{m} g(\dot{y}_j), \tag{2.6}$$

where the samples $\{\dot{y}_j; j = 1, \ldots, m\}$ are drawn from $p(y|x^{(n+1)}, D, M)$. If $\dot{\theta}_j$ is a sample from $p(\theta|D, M)$ and $\dot{y}_j$ is sample from $p(y|x^{(n+1)}, \dot{\theta}_j, D, M)$, then $\dot{y}_j$ is a sample from $p(y|x^{(n+1)}, D, M)$.

In the Markov chain Monte Carlo (MCMC) approach, samples are generated using a Markov chain that has the desired posterior distribution as its stationary distribution. The difficult part is to create a Markov chain that converges rapidly and in which the states visited after convergence are not highly dependent. Good introduction to basic MCMC methods and many applications in statistical data analysis can be found in (Gilks et al., 1996) and a more theoretical treatment in (Robert and Casella, 1999). Other excellent references discussing also various advanced methods are (Neal, 1993; Gamerman, 1997; Chen et al., 2000; Liu, 2001). Next, we mention the MCMC methods used in this work.

In the Metropolis-Hastings method (Metropolis et al., 1953; Hastings, 1970), the chain is constructed as follows:

1. Generate θ' from a proposal distribution $q(\theta|\theta^{(t)})$.
2. Compute
$$\alpha = \min\left(1, \frac{p(\theta', D, M)q(\theta^{(t)}|\theta')}{p(\theta^{(t)}, D, M)q(\theta'|\theta^{(t)})}\right). \tag{2.7}$$
3. Set $\theta^{(t+1)} = \theta'$ with probability α, otherwise $\theta^{(t+1)} = \theta^{(t)}$.

Its simplicity has made the basic Metropolis-Hastings algorithm one of the most used MCMC methods.

In the Gibbs sampling (Geman and Geman, 1984), each parameter is sampled in turn from the full conditional distribution of the parameter given all the other parameters and the data

$$p(\theta_k | \theta_{\setminus i}, D, M), \tag{2.8}$$

where $\theta_{\setminus i}$ denotes all the other parameters except θ_k. Gibbs sampling is useful if samples can easily be drawn from full conditional distribution, which is the case in many hierarchical models if the prior is chosen to be a conjugate distribution. Gibbs sampling is the main sampling method in the BUGS system (Gilks et al., 1992), which is a very convenient Bayesian modeling tool for experimenting with hierarchical Bayesian models.

The hybrid Monte Carlo (HMC) algorithm by Duane et al. (1987) is an elaborate Monte Carlo method, which makes efficient use of gradient information to reduce random walk behavior. The gradient indicates in which direction the algorithm should proceed to find high-probability states. The detailed description of the algorithm is not repeated here, see (Duane et al., 1987; Neal, 1993, 1996) instead. We have also used the windowed variation of HMC by Neal (1994, 1996).

Although the samples from the MCMC are dependent, approximation of Equation 2.6 is valid, but the variance estimates are trickier. To simplify computations (and save storage space), we have used thinning to get more independent MCMC samples (estimated by the autocorrelations (Neal, 1993, ch. 6; Chen et al., 2000, ch. 3)).

When the amount of data increases, the evidence from the data causes the probability mass to concentrate in a smaller area and thus we need fewer samples from the posterior distribution. In addition, fewer samples are needed to evaluate the mean of the predictive distribution than the tail-quantiles, such as the 10% and 90% quantiles. So depending on the problem, some hundreds of samples may be enough for practical purposes. Note that due to autocorrelations in the Markov chain, getting some 100 near-independent samples from a converged chain may require tens of thousands of samples from the chain, which may require several hours of CPU-time on a standard workstation.

For convergence diagnostics we have used visual inspection of trends, the potential scale reduction method (Gelman, 1996) and Kolmogorov-Smirnov test (Robert and Casella, 1999, ch. 8). Alternative convergence diagnostics have been reviewed, for example, in (Brooks and Roberts, 1999; Robert and Casella, 1999, ch. 8).

The MCMC sampling in our examples were done with the FBM software[1], and with Matlab code partially derived from the FBM and Netlab toolbox[2] .

[1] http://www.cs.toronto.edu/~radford/fbm.software.html

[2] http://www.ncrg.aston.ac.uk/netlab/

2.2 Models and priors

The probability model for the measurements, $p(y|x, \theta, D, M)$, contains the chosen approximation functions and noise models. It also defines the likelihood part in the posterior probability term, $p(\theta|D, M) \propto p(D|\theta, M)p(\theta|M)$. In a regression problem with additive error, the probability model is

$$y = f(x, \theta_w, D) + e, \tag{2.9}$$

where $f()$ is, for example, the MLP function, θ_w denotes the parameters of that function, and the random variable e is the model residual.

In two-class classification problems, the probability that a binary-valued target, y, has the value 1 may be computed by the logistic transformation (see, e.g., Bishop, 1995) as

$$p(y = 1|x, \theta_w) = [1 + \exp(-f(x, \theta_w))]^{-1}, \tag{2.10}$$

and in many-class classification problems the probability that a class target, y, has value j may be computed by the softmax transformation (or cross-entropy) (see, e.g., Bishop, 1995) as

$$p(y = j|x, \theta_w) = \frac{\exp(f_j(x, \theta_w))}{\sum_k \exp(f_k(x, \theta_w))}. \tag{2.11}$$

Next we discuss some useful residual models. A description of the MLP network (section 2.2.2) and Gaussian process (section 2.2.3) models, both of which are practical flexible nonlinear models, then follows. Finally, we discuss some important issues specific to input variable priors (section 2.2.4) and general prior sensitivity (section 2.2.5).

2.2.1 Residual models

The commonly used Gaussian noise model is

$$e \sim N(0, \sigma^2), \tag{2.12}$$

where $N(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2. In choosing the hyperprior for σ^2, there may be some prior knowledge allowing the use of a somewhat informative prior. For example, the minimum reasonable value for the noise variance can be estimated from measurement accuracy or from repeated experiments. Whether the hyperprior is informative or non-informative, it is convenient to choose the form of the distribution in accordance with the method used to sample from the posterior distribution. Note that the results in general are not very sensitive to the choices made at the hyperprior level, as discussed in section 2.1.3 and confirmed in many studies (see, e.g., Rasmussen, 1996). However, this should be checked in a serious analysis, especially if the form of the prior needs to be compromised for reasons of computational convenience. In the framework used in this study, the hyperparameters are sampled by Gibbs sampling. Convenient priors are thus conjugate distributions, which produce full conditional posteriors of the same form. For the variance of the Gaussian, a conjugate distribution is the inverse Gamma, producing the prior

$$\sigma^2 \sim \text{Inv-gamma}(\sigma_0^2, \nu_\sigma), \tag{2.13}$$

with the parametrization

$$\text{Inv-gamma}(\sigma^2|\sigma_0^2, \nu) \propto (\sigma^2)^{-(\nu/2+1)} \exp\left(-\frac{1}{2}\nu\sigma_0^2\sigma^{-2}\right),$$

which is equal to a scaled inverse chi-square distribution (Gelman et al., 1995, Appendix A). The parameter ν is the number of degrees of freedom and σ_0^2 is a scale parameter. In this parametrization, the prior is equivalent to having ν uncorrelated prior measurements with averaged squared deviation σ_0. The fixed values for σ_0 and ν_σ can be chosen to produce a vague prior for σ^2 that is reasonably flat over the range of parameter values that could plausibly arise. We have used $\sigma_0 = 0.05$ and $\nu_\sigma = 0.5$, similar to those used by Neal (1996, 1998).

Multivariate problems (with several outputs) can be handled by changing the output in Equation 2.9 to be a vector (thus having common residual model for all outputs), or by completely separate models, or as a hierarchical model with some common parts (that is, common hidden layer, separate output weights, and common or hierarchical noise model). In (Vehtari and Lampinen, 1999b) we analysed a multivariate regression problem where the residuals of the outputs may have been correlated. For a multivariate normal residual model with full covariance matrix, a conjugate hyperprior is the inverse Wishart distribution, allowing Gibbs sampling for the covariance matrix. See (Barnard et al., 2000) and references therein for alternative parametrizations for the covariance matrix.

In the noise model in Equation 2.12, the same noise variance σ^2 is assumed for each sample. In heteroscedastic regression problems, each sample $(x^{(i)}, y^{(i)})$ can have a different noise variance $(\sigma^2)^{(i)}$, with all the variances governed by a common prior, corresponding to, for example, a noise model

$$\begin{aligned} y^{(i)} &= f(x^{(i)}; \theta_w) + e^{(i)} \\ e^{(i)} &\sim N(0, (\sigma^2)^{(i)}) \\ (\sigma^2)^{(i)} &\sim \text{Inv-gamma}(\sigma_{\text{ave}^2}, \nu_\sigma) \\ \sigma_{\text{ave}}^2 &\sim \text{Inv-gamma}(\sigma_0^2, \nu_{\sigma,\text{ave}}), \end{aligned} \tag{2.14}$$

where the fixed hyperparameters are ν_σ, σ_0^2 and $\nu_{\sigma,\text{ave}}$. Here the prior spread of the variances $(\sigma^2)^{(i)}$ around the average variance σ_{ave}^2, determined by ν_σ, is fixed.

In this parametrization, the residual model is asymptotically equal to Student's t-distribution with fixed degrees of freedom. To allow for a higher probability for models with similar noise variances, the hyperparameter ν_σ can also be given a hyperprior, so that models with similar variances can have a large ν_σ, corresponding to a tight prior for the spread of variances $(\sigma^2)^{(i)}$. This is asymptotically the same as the t-distribution noise model with unknown degrees of freedom (Geweke, 1993). Thus, a similar treatment results, whether we assume normal residuals with different variances, or a common longer tailed t-distribution residual model, which is discussed in more detail below.

In heteroscedastic problems, the noise variance can be functionally dependent on some explanatory variables, typically on some subset of the model inputs, so that the model for the noise variance might be

$$\begin{aligned}(\sigma^2)^{(i)} &= F(x^{(i)}; \theta_{\text{noise}}) + \epsilon \\ \epsilon &\sim \text{Inv-gamma}(\sigma_0^2, \nu_\sigma)\end{aligned} \tag{2.15}$$

with fixed σ_0^2 and ν_σ. See (Bishop and Qazaz, 1997; Goldberg et al., 1998) for examples of input dependent noise models, where a separate MLP or GP model is used to estimate the dependence of the noise variance on the inputs.

Often the Gaussian residual model is not applicable in practical problems. There may be error sources that have non-Gaussian density, or the target function may contain peaks, but the training data is not sufficient to estimate them, or the data is heteroscedastic, with different noise variances in each sample. With a Gaussian residual model, samples with exceptionally large residuals must be handled as outliers, using pre- and postfiltering, and manual manipulation of data. Better option is to use a longer-tailed residual model that allows a small portion of samples to have large errors. An often-used model is the Laplace (or double exponential) distribution (Laplace, 1774). When the appropriate form for the residual distribution is not known in advance, the correct Bayesian treatment is to integrate over all *a priori* plausible forms.

In this study, we have used Student's t-distribution, where the tails can be controlled by choosing the number of degrees of freedom ν in the distribution. As this number is difficult to guess in advance, we set a hierarchical prior for it, and in the prediction we integrate over the posterior distribution given the data. Thus the tails are determined by the fit of the model to the data. The integration over the degrees of freedom can be done, for example, by Gibbs sampling (see section 2.1.5) for discretized values of ν (Spiegelhalter et al., 1996), so that the residual model is

$$
\begin{aligned}
e &\sim t_\nu(0, \sigma^2) \\
\sigma^2 &\sim \text{Inv-gamma}(\sigma_0, \nu_\sigma) \\
\nu &= V[i] \qquad (2.16) \\
i &\sim U_d(1, K) \\
V[1\!:\!K] &= [2,\ 2.3,\ 2.6,\ 3,\ 3.5,\ 4,\ 4.5,\ 5\!:\!1\!:\!10,\ 12\!:\!2\!:\!20,\ 25\!:\!5\!:\!50],
\end{aligned}
$$

where $[a\!:\!s\!:\!b]$ denotes the set of values from a to b with step s, and $U_d(a, b)$ is a uniform distribution of integer values between a and b. The discretization is chosen so that an equal prior for each value results in a roughly log-uniform prior on ν. Another simple way to sample ν, without discretization, is with the Metropolis-Hastings algorithm, which in our experiments with MLP gave equal results but slightly slower convergence. In the case of Gaussian process implementation, it is simpler to use a per-case variances model, that is, the heteroscedastic model of Equation 2.14 (see implementation details in Neal, 1997, 1999).

2.2.2 MLP neural networks

For MLP neural networks, the Bayesian approach was pioneered by Buntine and Weigend (1991), MacKay (1992), and Neal (1992) and reviewed, for example, by MacKay (1995), Neal (1996), Bishop (1995), and Lampinen and Vehtari (2001).

We used an MLP with a single hidden layer and tanh (hyperbolic tangent) hidden units, which in matrix format can be written as

$$f(x, \theta_w) = b_2 + w_2 \tanh\left(b_1 + w_1 x\right), \qquad (2.17)$$

where θ_w denotes all the parameters w_1, b_1, w_2, b_2, which are the hidden layer weights and biases, and the output layer weights and biases, respectively.

Typical prior assumptions in regularization theory are related to the smoothness of the approximation. In Tikhonov regularization (Bishop, 1995), which is a widely used regularization method in, for example, inverse problems, functions with large derivatives of chosen order are penalized. With an MLP model, minimizing the curvature (second derivative) (Bishop, 1993) or training the derivatives to given target values (Lampinen and Selonen, 1997) leads to a rather complex treatise as the partial derivatives of the nonlinear models depend on all the other inputs and weights.

A convenient commonly used prior distribution is the Gaussian, which in linear models is directly related to model derivatives, but has a more complex interpretation in the nonlinear MLP case, as discussed in section 2.2.4. The Gaussian priors for the weights are

$$w_1 \sim N(0, \alpha_{w_1}) \tag{2.18}$$

$$b_1 \sim N(0, \alpha_{b_1}) \tag{2.19}$$

$$w_2 \sim N(0, \alpha_{w_2}) \tag{2.20}$$

$$b_2 \sim N(0, \alpha_{b_2}), \tag{2.21}$$

where the α's are the variance hyperparameters. The conjugate inverse Gamma hyperprior is

$$\alpha_j \sim \text{Inv-gamma}(\alpha_{0,j}, \nu_{\alpha,j}) \tag{2.22}$$

similarly as for the hyperpriors in the Gaussian noise model. The fixed values for the highest-level hyperparameters in the case studies were similar to those used by Neal (1996, 1998). The appropriate hyperpriors depend somewhat on the network topology. As discussed by Neal (1996) the average weights can be assumed to be smaller when there are more feeding units, thus for example, the hyperprior for w_1 is scaled according to the number of inputs K. Typical values used were

$$\begin{aligned} \nu_{\alpha,w_1} &= 0.5 \\ \alpha_{0,w_1} &= (0.05/K^{1/\nu_{\alpha,w_1}})^2. \end{aligned} \tag{2.23}$$

We have also used a simple hierarchical prior for the MLP weights, called Automatic Relevance Determination (ARD) (MacKay, 1994; Neal, 1996, 1998). In ARD each group of weights connected to the same input $k \in \{1, \ldots, K\}$ has common variance hyperparameters, while the weight groups can have different hyperparameters. An example of the ARD prior used in this study is

$$w_{kj} \sim N(0, \alpha_k), \tag{2.24}$$

$$\alpha_k \sim \text{Inv-gamma}(\alpha_{\text{ave}}, \nu_\alpha) \tag{2.25}$$

$$\alpha_{\text{ave}} \sim \text{Inv-gamma}(\alpha_0, \nu_{\alpha,\text{ave}}), \tag{2.26}$$

where the average scale of the α_k is determined by the next level hyperparameters, in similar fashion as in the heteroscedastic noise model example above. The fixed values used in the case studies were $\nu_\alpha = 0.5$, $\alpha_0 = (0.05/K^{1/\nu_\alpha})^2$ and $\nu_{\alpha,\text{ave}} = 1$, corresponding to vague hyperpriors that let α_{ave} and α_k be determined by the data. The hyperparameter ν_α could also be given a hyperprior in similar fashion as in the heteroscedastic noise model example. In section 2.2.4 we discuss ARD in more detail.

In the framework introduced by Neal (1996), the hybrid Monte Carlo algorithm is used for sampling the parameters and Gibbs sampling for the hyperparameters. The detailed description of the algorithm is not repeated here, see instead (Neal, 1996). For other possible sampling schemes see, for example, (Müller and Rios Insua, 1998; de Freitas et al., 2000).

In (Vehtari et al., 2000) we discussed the choice of the starting values and the number of chains. Choosing the initial values with early-stopping (Morgan and Bourland, 1990; Prechelt, 1998) can be used to reduce the burn-in time when the chain has not yet reached the equilibrium distribution. In general, the author's experience suggests that the convergence of the MCMC methods for MLP is slower than usually assumed, so that in many of the published studies, the MCMC chains may have still been in burn-in stage, producing a sort of early-stopping effect to the selection of the model complexity.

The MLP architecture causes some MCMC sampling problems, due to many correlated parameters and possible posterior multimodality (Neal, 1996; Müller and Rios Insua, 1998; Vehtari et al., 2000). Additionally, it is not entirely clear what are the properties of the realized prior on functions when having a finite number of hidden units and a Gaussian prior on weights. Nevertheless, the MLP is useful model as it is reasonable efficient even with large data sets.

2.2.3 Gaussian processes

The Gaussian process is a non-parametric regression method, with priors imposed directly on the covariance function of the resulting approximation (see, e.g., Rasmussen, 1996; Williams and Rasmussen, 1996; Gibbs, 1997; Barber and Williams, 1997; Neal, 1997; MacKay, 1998a; Neal, 1999).

Given the training inputs $x^{(1)}, \dots, x^{(n)}$ and the new input $x^{(n+1)}$, a covariance function can be used to compute the $n+1$ by $n+1$ covariance matrix of the associated targets $y^{(1)}, \dots, y^{(n)}, y^{(n+1)}$. The predictive distribution for $y^{(n+1)}$ is obtained by conditioning on the known targets, giving a Gaussian distribution with the mean and the variance given by

$$E_y[y|x^{(n+1)}, \theta, D] = k^{\mathrm{T}} C^{-1} y^{(1,\dots,n)} \tag{2.27}$$

$$\mathrm{Var}_y[y|x^{(n+1)}, \theta, D] = V - k^{\mathrm{T}} C^{-1} k, \tag{2.28}$$

where C is the n by n covariance matrix of the observed targets, $y^{(1,\dots,n)}$ is the vector of known values for these targets, k is the vector of covariances between $y^{(n+1)}$ and the known n targets, and V is the prior variance of $y^{(n+1)}$. For regression, we used a simple covariance function producing smooth functions

$$C_{ij} = \eta^2 \exp\left(- \sum_{u=1}^{p} \rho_u^2 (x_u^{(i)} - x_u^{(j)})^2 \right) + \delta_{ij} J^2 + \delta_{ij} \sigma_e^2. \tag{2.29}$$

The first term of this covariance function expresses that the cases with nearby inputs should have highly correlated outputs. The η parameter gives the overall scale of the local correlations. The ρ_u parameters are multiplied by the coordinate-wise distances in input space and thus allow for different distance measures for each input dimension. The second term is the jitter term, where $\delta_{ij} = 1$ when

$i = j$. It is used to improve matrix computations by adding a constant term to residual model. The third term is the residual model.

We used an Inverse-gamma prior on η^2 and a hierarchical Inverse-gamma prior (producing an ARD like prior) on ρ_u.

$$\begin{aligned} \eta^2 &\sim \text{Inv-gamma}(\eta_0^2, \nu_{eta^2}) \\ \rho_u &\sim \text{Inv-gamma}(\rho_{\text{ave}}, \nu_\rho) \\ \rho_{\text{ave}} &\sim \text{Inv-gamma}(\rho_0, \nu_0) \\ \nu_\rho &\sim \text{Inv-gamma}(\nu_{\rho,0}, \nu_{\nu_\rho,0}). \end{aligned} \tag{2.30}$$

In the framework introduced by Neal (1997, 1999), the parameters of the covariance function are sampled using the HMC, and the per-case variances are sampled with Gibbs sampling. The detailed description of the algorithm is not repeated here, see instead (Neal, 1997, 1999).

The GP may have much less parameters than the MLP and a part of the integration is done analytically, and thus the MCMC sampling may be much easier than for the MLP. On the other hand, there may be memory and performance problems when the sample size is large and sampling of many latent values may be slow. Memory and performance problems arise because the approach used requires inversion of n times n matrix, which is an $O(n^3)$ operation. There are alternative approaches, which can approximate the matrix inversion by sacrificing flexibility and/or accuracy (Gibbs, 1997; Zhu et al., 1998; Trecate et al., 1999; Csató et al., 2000; Williams and Seeger, 2001; Smola and Bartlett, 2001). However, GP is very viable alternative for MLP, at least in problems in which the training sample size is not very large.

2.2.4 The automatic relevance determination prior

The ARD prior was proposed by MacKay (1994) and Neal (1996) as an automatic method for determining the relevance of the inputs in MLP, as irrelevant inputs should have smaller weights in the connections to the hidden units than more important weights. With separate hyperparameters, the weights from irrelevant inputs can have tighter priors, which reduce such weights more effectively towards zero than having a common larger variance for all the input weights.

Determining the relevance of inputs has a great importance in practical modeling problems, in both choosing the inputs in the models as well as in analyzing the final model. See (Sarle, 1997) for a general discussion on ways to assess the importance of inputs in nonlinear models. The most common notions of importance are predictive importance (the increase in generalization error if the variable is omitted from the model) and causal importance (the change of model outputs caused by the change of input variable). Note that causal importance is directly measurable only if the inputs are uncorrelated (so that inputs can be manipulated independently), and that it is not related to the causality relations in the actual system to be modeled.

In ARD for the MLP, the relevance measure of an input is related to the size of the weights connected to that input. In linear models, these weights define the partial derivatives of the output with respect to the inputs, which is equal to the predictive importance of the input, and in the case of non-correlated inputs, also to the causal importance. In nonlinear MLP networks the situation is, however, more complex, since small weights in the first layer can be compensated by large weights in other layers, and the nonlinearity in the hidden units changes the effect of the input in a way that depends on all the other inputs.

To illustrate the effect of an ARD prior, consider a K-J-1 MLP with a linear output layer,

$$y = \sum_{j=1}^{J} v_j S\left(\sum_{k=1}^{K} w_{kj} x_k\right). \tag{2.31}$$

The d'th order partial derivative of the mapping is

$$\frac{\partial^d y}{\partial (x_k)^d} = \sum_j v_j (w_{kj})^d S^{(d)}\left(\sum_k w_{kj} x_k\right), \tag{2.32}$$

where $S^{(d)}$ is the d'th derivative of S. Thus, constraining the first layer weights has the largest effect on the higher order derivatives, in the d'th order polynomial term $(w_{kj})^d$. This may partly explain the success of weight decay regularization, as this type of prior is an effective smoothing prior. On the other hand, to produce a linear mapping with small high order derivatives, the first layer weights would need to be small, so that the sigmoids operate on the linear part, and the second layer weights correspondingly larger. Thus the first layer weights do not measure the first derivative, or the linear relation, no matter how important it is. The network may also contain direct input-to-output weights to account for any linear relation (see, e.g., Neal, 1996), but the ARD coefficients of these weights are not comparable to the ARD coefficients of the hidden layer weights. Note that adding input-to-output weights makes the model less identifiable and may slow down the convergence of MCMC considerably (Neal, 1998).

Similar to MLP, in GP the "ARD" parameters ρ_u measure the nonlinearity of the inputs. The parameter ρ_u defines the characteristic length of the function for given input direction. The characteristic length describes the length where substantial changes in the function may happen. Both irrelevant inputs and inputs with near linear effect have long characteristic length, except in the case of important inputs, in which characteristic length is limited by the range of the data.

In the following simple example, we demonstrate how the nonlinearity of the input has the largest effect on the relevance score of the ARD in MLP, instead of

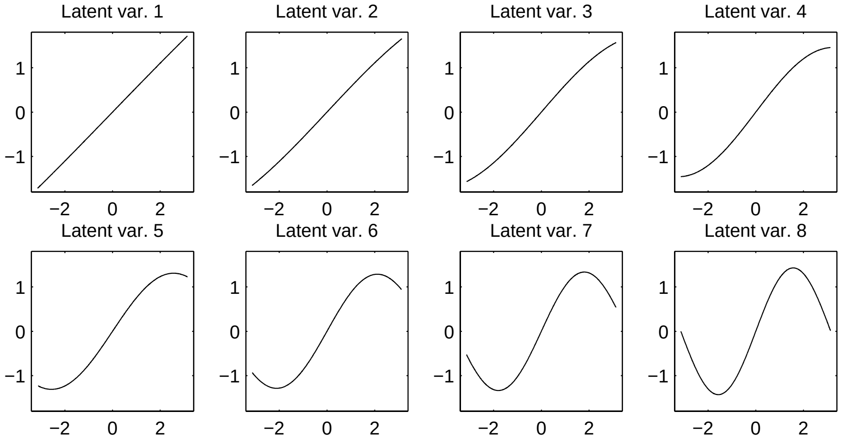

Figure 2.1: An example of ARD and the importance of inputs. The target function is an additive function of eight inputs. The plots show the univariate transformations of the inputs. The predictive importance of every input is equal, in RMSE terms, as the latent functions are scaled to equal variance over the uniform input distribution $U(-3, 3)$.

the predictive or causal importance. The target function is an additive function of eight inputs (see Figure 2.1), with equal predictive importance for every input. The network weights (using the evidence approximation) are shown in Figure 2.2, from where it is easy to see how the weights connected to the inputs with linear transformation are smallest. Figure 2.3 shows the predictive importance and the mean absolute values of the first and second order derivatives of the output with respect to each input, and the relevance estimates from the ARD (posterior standard deviation of the Gaussian prior distributions for each weight group). The example illustrates how the inputs with a large but linear effect are given low relevance measures by the ARD. For this reason one should be cautious of using the ARD to choose or remove inputs in the models, or to rank the variables according to importance in the analysis of the model. In section 4.3.1 we illustrate the difference between the ARD values of MLP and GP, and the marginal posterior probabilities of the inputs, and demonstrate how ARD under-estimates the predictive relevance of near linear input.

Note however that ARD is often a very favorable prior, as demonstrated in the case studies in this work, since it loosens the more strict assumption that all the input weight groups should have the same variance (or nonlinearity). So, unless the variance is actually assumed to be the same, ARD should be used as a less informative but probably more correct prior.

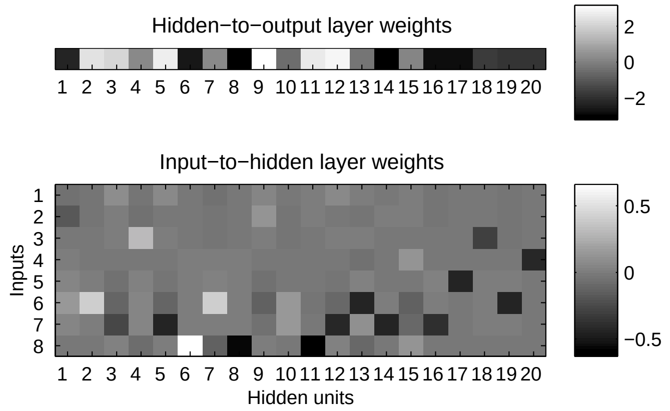

Figure 2.2: Network weights for the test function in Figure 2.1, estimated using the evidence framework.

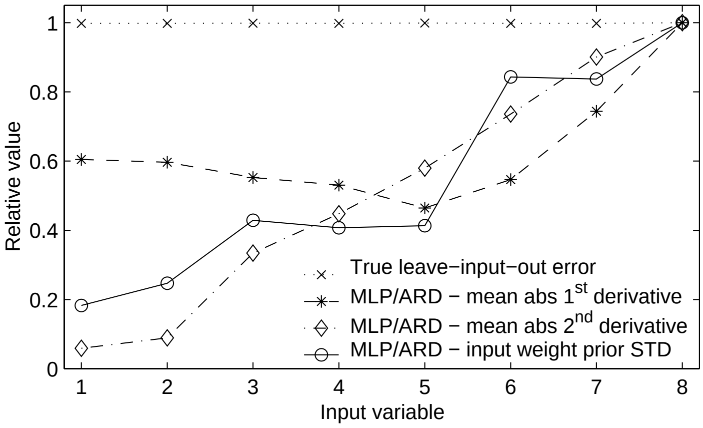

Figure 2.3: Different measures of importance of inputs for the test function in Figure 2.1. Note, that the ARD coefficients are closer to the second derivatives than to the first derivatives (local causal importance) or to the error due to leaving input out (predictive importance).

2.2.5 On sensitivity to the prior distributions

As explained above, the Bayesian approach is based on averaging probable models, where the probability is computed from the chosen distributions for the noise models, parameters etc. Thus, the approach may be more sensitive to bad guesses for these distributions than more classical methods, in which model selection is carried out as an external procedure, such as cross-validation that is based on fewer assumptions (mainly on the assumption that the training and validation sets are not correlated). In this respect, Bayesian models can also be overfitted in terms of classical model fitting, to produce too complex models and too small posterior estimates for the noise variance. To check the assumptions of the Bayesian models, we always carry out the modeling with simple classical methods (such as linear models, early-stopped committees of MLPs, etc.). If the Bayesian model gives inferior results (measured with an independent test set or with cross-validation), some of the assumptions are questionable. The prior sensitivity also appears in prior-predictive densities and Bayes factors discussed in section 3.3.1.

The following computer simulation illustrates the sensitivity of the Bayesian approach to the correctness of the noise model, compared to the early-stopped committee (ESC), which is a robust reference method used in all our case studies. In early stopping (Morgan and Bourland, 1990; Prechelt, 1998) weights are initialized to very small values. Part of the training data is used to train the MLP and the other part is used to monitor the validation error. Iterative optimization algorithms used for minimizing the training error gradually take parameters in use. Training is stopped when the validation error begins to increase. The basic early stopping is statistically rather inefficient, as it is very sensitive to the initial conditions of the weights and only a part of the available data is used to train the model. These limitations can easily be alleviated by using a committee of early stopping MLPs, with a different partitioning of the data to training and stopping sets for each MLP (Krogh and Vedelsby, 1995). When used with caution, the early stopping committee is a good baseline method for MLPs.

The target function and data are shown in Figure 2.4. The modeling test was repeated 100 times with different realizations of Gaussian or Laplacian (double exponential) noise. The model was a 1-10-1 MLP with a Gaussian noise model. The figure shows one realization of the data and the resulting predictions. The 90% error intervals, or credible intervals (CI), are for the predicted conditional mean of the output given the input, thus the measurement noise is not included in the limits. For the ESC, the intervals are simply computed separately for each x-value from 100 networks. Computing the confidence limits for early-stopped committees is not straightforward, but this very simple *ad hoc* method often gives results similar to the Bayesian MLP treatment. The summary of the experiment is shown in Table 2.1. Using the paired t-test, the ESC is significantly better than the Bayesian model when the noise model is wrong. In this simple problem, both

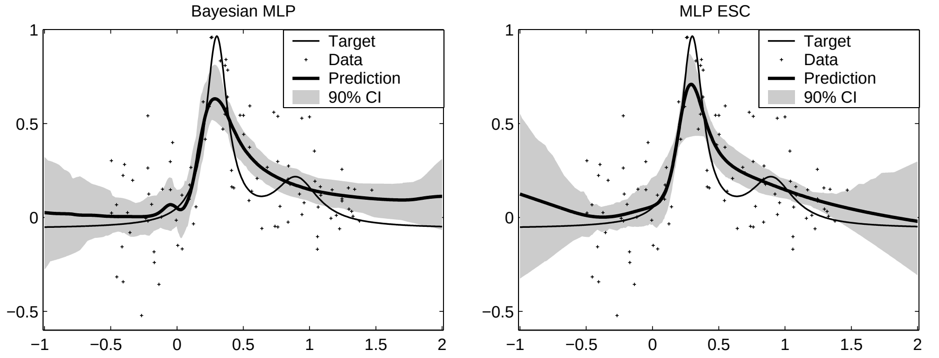

Figure 2.4: The test function in demonstrating the sensitivity of a Bayesian MLP and an early-stopped committee (MLP ESC) to the wrong noise model. The figure shows one realization of the data and the resulting predictions, with Bayesian MLP in the left and MLP ESC in the right figure. See text for explanation for the credible intervals (CI).

Table 2.1: Demonstration of the sensitivity of Bayesian MLP and MLP ESC to wrong noise model. For both models, the noise model was Gaussian, and the actual noise Gaussian or Laplacian (double exponential). The statistical significance of the difference is tested with the paired t-test. The errors are RMS errors of the prediction with respect to the true target function.

Noise	Bayesian MLP RMSE	MLP ESC RMSE	Probability of the difference
Gaussian	0.2779	0.2784	0.15
Laplacian	0.2828	0.2766	0.99

methods are equal for the correct noise model. The correct Bayesian approach of integrating over the noise models, as explained in section 2.1.3 and shown in practice in a case problem in section 2.2.1, would of course have no trouble in this example.

The implication of this issue in practical applications is that the Bayesian approach usually requires more expert work than other approaches, either to devise reasonable assumptions for the distributions, or to include different options in the models and integrate over them, but that done, in our experience the results are systematically better than with other approaches.

2.3 Illustrative examples

As illustrative examples we use MLP networks and Gaussian processes with Markov Chain Monte Carlo sampling in three real world problems: concrete quality estimation (section 2.3.1), forest scene classification (section 2.3.2) and electrical impedance tomography (section 2.3.3).

2.3.1 Case I: Regression task in concrete quality estimation

In this section we present results from the real world problem of predicting the quality properties of concrete (first analysed in Vehtari and Lampinen, 1999a).

The goal of the project was to develop a model for predicting the quality properties of concrete, as a part of a large quality control program of the industrial partner of the project. The quality variables included for example compressive strengths and densities for 1, 28 and 91 days after casting, and bleeding (water extraction), flow value, slump and air-%, which measure the properties of fresh concrete. These quality measurements depend on the properties of the stone material (natural or crushed, size and shape distributions of the grains, mineralogical composition), additives, and the amount of cement and water. In the study we had 27 explanatory variables selected by the concrete expert, (listed for example in Figure 4.3) and 215 samples designed to cover the practical range of the variables, collected by the concrete manufacturing company. The details of the problem, the descriptions of the variables and the conclusions made by the concrete expert are given in (Järvenpää, 2001).

Collecting the samples for statistical modeling is rather expensive in this application, as each sample requires preparation of the sand mixture, casting the test pieces and waiting 91 days for the final tests. In small sample problems, the selection of correct model complexity is more important and needs to be done with finer resolution than in problems with large amounts of data. This makes hierarchical Bayesian models a tempting alternative.

In the study we used 10-hidden-unit MLP networks and a Gaussian process. To illustrate the effect of priors, four different MLPs with different priors were tested. As a reference method, we used an early stopping committee of 10 MLP networks, with different division of data into training and stopping sets for each member. The networks were initialized to near zero weights to guarantee that the mapping was smooth in the beginning.

In the following we report the results for one variable, air-%, which measures the volume percentage of air in the concrete. As the air-% is positive and has a very skewed distribution (with mean 2.4% and median 1.7%), we use logarithmic transformation for the variable. This ensures the positiveness and allows the use of much simpler additive noise models than in the case of a nearly exponentially distributed variable.

Table 2.2: Performance comparison of various MLP models and a Gaussian process model in predicting the air-% variable in concrete manufacturing. The presented RMSE values show the standardized model residuals relative to the standard deviation of the data.

Method		Noise model	ARD	RMSE	std
1.	MLP ESC	N		0.30	0.04
2.	Bayesian MLP	N		0.26	0.04
3.	Bayesian MLP	t_ν		0.24	0.03
4.	Bayesian MLP	N	yes	0.21	0.02
5.	Bayesian MLP	t_ν	yes	0.19	0.02
6.	Gaussian process	t_ν	yes	0.19	0.02

The performance of the models was estimated by 10-fold cross-validation and the model comparison was made using a paired t-test. Although this approach is not as accurate as approach described in Chapter 3, this is a valid approximation of the expected utility approach, and adequate for these comparisons.

The estimated prediction errors are presented in the Table 2.2. In the column *Noise model* the letter N indicates normal noise model and t_ν Student's t-distribution with an unknown number of degrees of freedom ν, respectively.

The posterior values for ν were, for example, for a Bayesian MLP with ARD prior, between 2 and 3.5 (10% and 90% quantiles), corresponding to rather long tailed distribution for the model residuals.

Some conclusions from the results are listed in the following. The best models, the Gaussian process model and the Bayesian MLP, with ARD and Student's t_ν-distribution with an unknown number of degrees of freedom as noise model, performed equally well. The best models were those with the most flexible (less informative) priors. Within the MLP models, the t_ν noise model outperformed all the Gaussian noise models with a 98% probability, and the MLP with t_ν but without ARD with a 93% probability. The early-stopped committee MLP ESC and the basic Bayesian MLP with the Gaussian noise model did not differ significantly. Just adding the Bayesian treatment for the basic model does not help in this application, if the possibility for using hierarchical priors is not utilized. Adding the ARD made the Bayesian MLP significantly better (99% probability) than the MLP ESC. Using just the longer tail noise model t_ν without ARD, made the Bayesian model better than the ESC MLP with 95% probability.

In the review above, we have presented the results for only one variable in the study. Rather similar results were obtained for the other variables. The Bayesian MLP and the Gaussian process model had very similar performance for all the target variables, so that the choice between them in this application is a matter of convenience. In Chapter 3 we take a more accurate look into comparing different residual models using the expected utilities approach.

2.3.2 Case II: Classification task in forest scene analysis

In this section, we present results from the real world problem of classification of forest scenes with MLP (first analysed in Vehtari et al., 1998).

The final objective of the project was to assess the accuracy of estimating the volumes of growing trees from digital images. To locate the tree trunks and to initialize the fitting of the trunk contour model a classification of the image pixels to tree and non-tree classes was necessary. The main problem in the task was the large variance in the classes. The appearance of the tree trunks varies in color and texture due to varying lighting conditions, epiphytes (such as gray or black lichen on white birch), and species dependent variations (such as the Scotch pine, with bark color ranging from dark brown to orange). In the non-tree class the diversity is much larger, containing, for example, terrain, tree branches, and sky. This diversity makes it difficult to choose the optimal features for the classification.

We extracted 84 potentially useful features: 48 Gabor filters (with different orientations and frequencies), which are generic features related to shape and texture, and 36 common statistical features (mean, variance and skewness with different window sizes), see details in (Vehtari et al., 1998).

Due to the large number of features, many classifier methods suffer from the curse of dimensionality. The results of this case demonstrate that the Bayesian MLP is very competitive in this high dimensional problem.

A total of 48 images were collected using an ordinary digital camera in varying weather conditions. The labeling of the image data was done by hand via identifying many types of tree and background image blocks with different textures and lighting conditions. In this study only pines were considered.

To estimate the classification errors of different models we used the eight-fold cross-validation error estimate, that is, 42 of 48 pictures were used for training and the six left out for error evaluation, and this scheme was repeated eight times.

The Gaussian process implementation discussed in section 2.2.3 requires the inversion of a n times n matrices many times, which in the case of 4800 data points was computationally infeasible. However, MLP networks work well in this problem, as their computation time scales linearly to n. We used a 20-hidden-unit MLP with the logistic likelihood model. The other tested models were:

- KNN LOO-CV, K-nearest-neighbor classification, where K is chosen by leave-one-out cross-validation[3], and
- CART, Classification And Regression Tree (Breiman et al., 1984).

The CV error estimates are collected in Table 2.3. The differences are not very significant because the different images had very different error rates. This causes extra variance to the classification results during the CV, which reduces the significance of the differences, even though the variance comes from the variations in the task, not variations of the models. See the more detailed analysis in Chapter 3.

[3] http://www.cs.utoronto.ca/˜delve/methods/knn-class-1/home.html

Table 2.3: CV error estimates for forest scene classification. See text for explanation of the different models.

Reference model	Classification error %	std
1. CART	30	2
2. KNN LOO-CV	20	2
3. MLP ESC	13	1
4. Bayesian MLP	12	1
5. Bayesian MLP + ARD	11	1

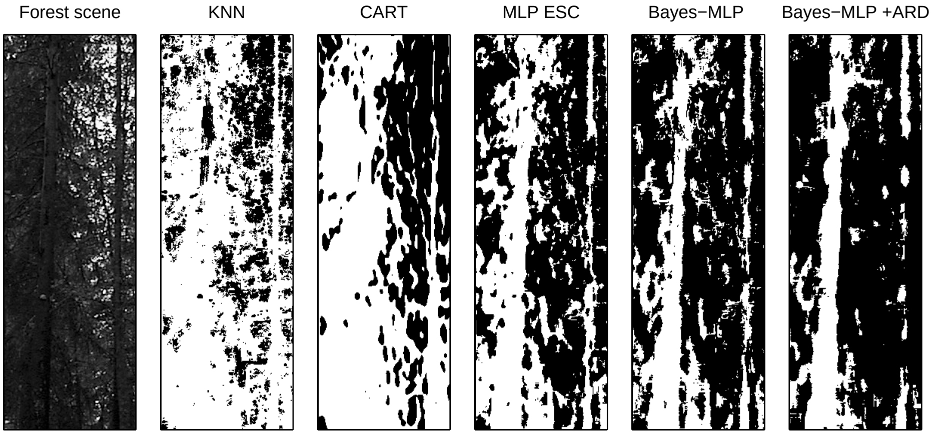

Figure 2.5: Examples of a classified forest scene. See text for explanation of the different models.

All the MLP models clearly outperform the other models, while the best model, the Bayesian MLP with ARD, is just slightly better than the other MLP models.

Figure 2.5 shows an example of a new unseen image classified with different models. Visually, the Bayesian MLP with ARD gives less spurious detections than the other models. The ARD reduces the effect of features correlating weakly with the classification, and thus larger windows and robust features dominate. On the other hand, this causes the classifier to miss some thin trunks and parts of trunks that are not clearly visible.

2.3.3 Case III: Inverse problem in electrical impedance tomography

In this section we report results on using Bayesian MLPs for solving an ill-posed inverse problem in electrical impedance tomography (EIT). The full report of the proposed approach is presented in (Lampinen et al., 1999; Vehtari and Lampinen, 2000).

The aim of EIT is to recover the internal structure of an object from surface measurements. A number of electrodes are attached to the surface of the object, current patterns are injected through the electrodes, and the resulting potentials are measured. The inverse problem in EIT, estimating the conductivity distribution from the surface potentials, is known to be severely ill-posed, and thus some regularization methods must be used to obtain feasible results (see, e.g., Kaipio et al., 2000).

Figure 2.6 shows a simulated example of the EIT problem. The volume bounded by the circles in the image represents a gas bubble floating in liquid. The conductance of the gas is much lower than that of the liquid, producing the equipotential curves shown in the figure. Figure 2.7 shows the resulting potential signals, from which the image is to be recovered.

In (Lampinen et al., 1999) we proposed a novel feedforward solution for the reconstruction problem. The approach is based on computing the principal component decomposition for the potential signals and the eigenimages of the bubble distribution from the autocorrelation model of the bubbles. The input to the MLP is the projection of the potential signals to the first principal components, and the MLP gives the coefficients for reconstructing the image as a weighted sum of the eigenimages. The projection of the potentials and the images to the eigenspace reduces correlations from the input and the output data of the network and detaches the actual inverse problem from the representation of the potential signals and image data.

The reconstruction was based on 20 principal components of the 128 dimensional potential signal and 30 eigenimages with a resolution of 41×41 pixels. The training data consisted of 500 simulated bubble formations with one to ten overlapping circular bubbles in each image. To compute the reconstructions, MLPs with 30 hidden units were used. Models tested were *MLP ESC* and *Bayesian MLP* (see section 2.3.1). Because of the input projection, ARD prior should not make much difference in results (this was verified in preliminary tests), and so a model with ARD prior was not used in full tests.

The reference method in the study was iterative inversion of the EIT forward model using total variation regularization (for further information see, e.g., (Kaipio et al., 2000)). In this approach, the conductivity distribution is sought to minimize a cost function, which is defined as the squared difference of the measured potentials and the potentials computed from the conductivity distribution by the forward model. The minimization was carried out by Newton's method, requiring

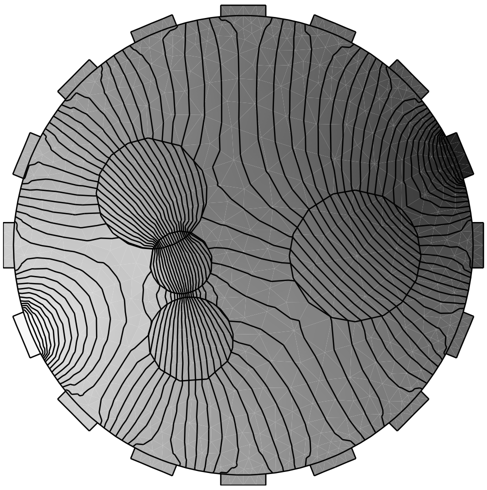

Figure 2.6: Example of the EIT measurement. The simulated bubble formation is bounded by the circles. The current is injected from the electrode with the lightest color and the opposite electrode is grounded. The gray level and the contour curves show the resulting potential field.

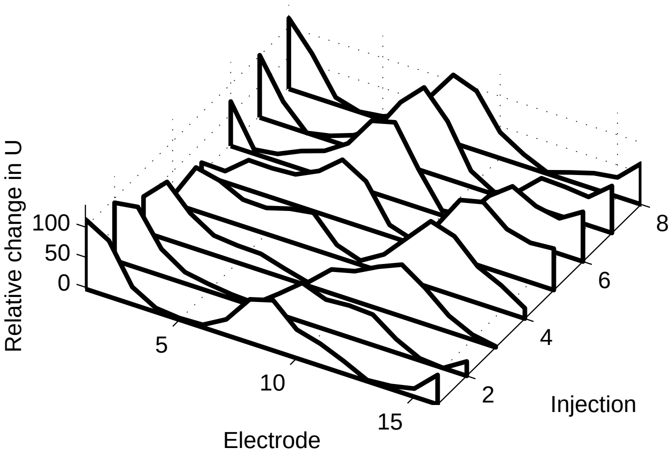

Figure 2.7: Relative changes in potentials compared to homogeneous background. The eight curves correspond to injections from eight different electrodes.

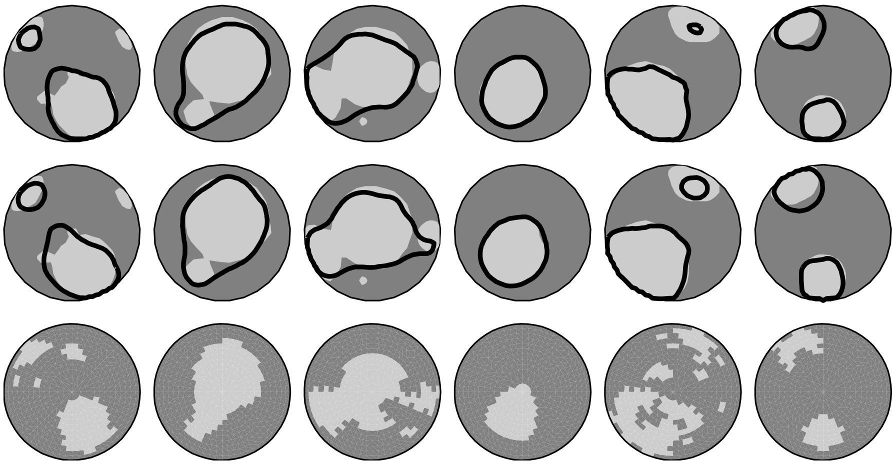

Figure 2.8: Example of image reconstructions with MLP ESC (upper row), Bayesian MLP (middle row), and TV inverse (lower row). In the MLP plots the actual bubble is shown by the gray blob and contour of the detected bubble as the black line. For the TV inverse, the estimated bubble is shown as the gray blob, with the same actual bubbles as in the upper images.

about 20 iteration steps. As it was known that the bubbles and the background had constant conductivities, total variation regularization was used. The regularizer penalty function was the total sum of absolute differences between adjacent area elements, forcing the solution to be smoother, but not penalizing abrupt changes (total change in a monotonic curve is equal independent of the steepness, in contrast to, say, squared differences that pull the solution towards low-gradient solutions).

Figure 2.8 shows examples of the image reconstruction results. Table 2.4 shows the quality of the image reconstructions, measured by the error in the void fraction and the percentage of erroneous pixels in the segmentation, over the test set. An important goal in this process tomography application was to estimate the void fraction, which is the proportion of gas in the image. With the proposed approach, such goal variables can be estimated directly without explicit reconstruction of the image. The last column in Table 2.4 shows the relative absolute error in estimating the void fraction directly from the projections of the potential signals.

In solving real problems, the ability to assess the confidence of the output is necessary. Figure 2.9 shows the scatter plot of the void fraction versus the estimate, together with credible intervals. The 10% and 90% quantiles are computed directly from the posterior predictive distribution of the model output (Equation 2.3). When the void fraction is large, the forward model becomes more

Table 2.4: Errors in reconstructing the bubble shape and estimating the void fraction from the reconstructed images. See text for explanation of the models.

Method	Classification error %	Relative error in VF %	Relative error in direct VF, %
TV-inverse	9.7	22.8	-
MLP ESC	6.7	8.7	3.8
Bayesian MLP	5.9	8.1	3.4

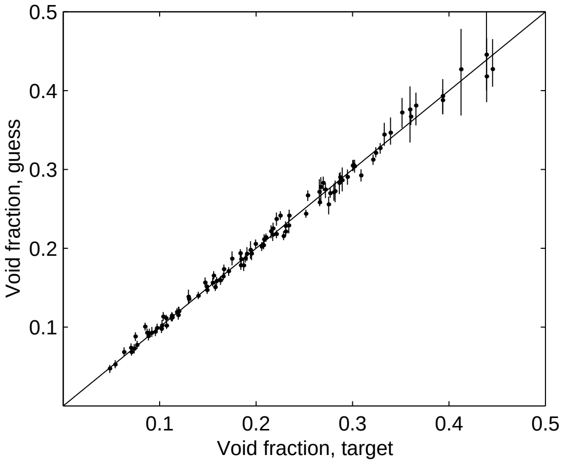

Figure 2.9: Scatterplot of the void fraction estimate with 10% and 90% quantiles.

nonlinear and the inverse problem becomes more ill-posed, especially for disturbances far from the electrodes. This ambiguity is clearly visible in the credible intervals, so that they are wider when the model may make large errors.

Although the model was based on simulated data, it has given very promising results in the preliminary experiments with real data.

Chapter 3

Model assessment and selection using expected utilities

3.1 Introduction

In this chapter we describe how to estimate the distributions of the expected utilities using cross-validation predictive densities. First we review expected utilities (section 3.1.1) and cross-validation predictive densities (section 3.1.2) and briefly discuss assumptions made on future data distribution in the approach described in this chapter and in related approaches where the goal is to compare (not assess) the performance of *methods* instead of the *models* (section 3.1.3).

We discuss the properties of two practical methods, the importance sampling leave-one-out (section 3.2.1) and the k-fold cross-validation (section 3.2.3). We propose a quick and generic approach based on the Bayesian bootstrap for obtaining samples from the distributions of the expected utilities (section 3.2.4). If there is a collection of models under consideration, the distributions of the expected utilities can also be used for comparison (section 3.2.5).

We also discuss the relation of the proposed method to prior predictive densities and Bayes factors (section 3.3.1), other predictive densities and respective Bayes factors (sections 3.3.2 and 3.3.3), and information criteria and the effective number of parameters (section 3.3.4).

As the estimation of the expected utilities requires a full model fitting (or k model fittings) for each model candidate, the proposed approach is useful only when selecting between a few models. If we have many model candidates, for example if doing variable selection, we can use some other methods like the variable dimension MCMC methods (Green, 1995; Carlin and Chib, 1995; Stephens, 2000) for model selection and still use the expected utilities for final model assessment. This approach is discussed in more detail in Chapter 4.

To illustrate the discussion we use MLP networks and Gaussian processes in one toy problem and two real world problems (section 3.4).

3.1.1 Expected utilities

In prediction and decision problems, it is natural to assess the predictive ability of the model by estimating the expected utilities (Good, 1952; Bernardo and Smith, 1994). Utility measures the relative values of consequences. By using application specific utilities, the expected benefit or cost of using the model for predictions or decisions (e.g., by financial criteria) can be readily computed. In lack of application specific utilities, many general discrepancy and likelihood utilities can be used.

The posterior predictive distribution of output y for the new input $x^{(n+1)}$ given the training data $D = \{(x^{(i)}, y^{(i)}); i = 1, 2, \ldots, n\}$ is obtained by integrating the predictions of the model with respect to the posterior distribution of the model (see section 2.1.2)

$$p(y|x^{(n+1)}, D, M) = \int p(y|x^{(n+1)}, \theta, D, M) p(\theta|D, M) d\theta. \tag{3.1}$$

We would like to estimate how good our model is by estimating how good predictions (i.e., the predictive distributions) the model makes for future observations from the same process that generated the given set of training data D. The goodness of the predictive distribution $p(y|x^{(n+h)}, D, M)$ can be measured by comparing it to the actual observation $y^{(n+h)}$ with the utility u

$$u_h = u\left(y^{(n+h)}, x^{(n+h)}, D, M\right). \tag{3.2}$$

The goodness of the whole model can then be summarized by computing some summary quantity of distribution of u_h's over all future samples ($h = 1, 2, \ldots$), for example, the mean

$$\bar{u} = E_h[u_h] \tag{3.3}$$

or an α-quantile

$$\bar{u}_\alpha = Q_{\alpha,h}[u_h]. \tag{3.4}$$

We call all such summary quantities the expected utilities of the model. Note that, considering the expected utility just for the next sample (or single one time decision) and taking the expectation over the distribution of $x^{(n+1)}$ is equivalent to taking the expectation over all future samples.

Preferably, the utility u would be application specific, measuring the expected benefit or cost of using the model. For simplicity, we mention here some general utilities. Both the square error

$$u_h = \left(E_y[y|x^{(n+h)}, D, M] - y^{(n+h)}\right)^2 \tag{3.5}$$

and the absolute error

$$u_h = \text{abs}\left(E_y[y|x^{(n+h)}, D, M] - y^{(n+h)}\right) \tag{3.6}$$

measure the accuracy of the expectation of the predictive distribution, but the absolute error is more easily understandable especially when summarized using α-quantile (e.g., $\alpha = 90\%$) as most of the predictions will have error less than the given value. The predictive likelihood measures how well the model models the predictive distribution

$$u_h = p(y^{(n+h)} | x^{(n+h)}, D, M) \tag{3.7}$$

and it is especially useful in model comparison (see section 3.2.5) and in non-prediction problems. Maximization of the expected predictive likelihood corresponds to minimization of information-theoretic Kullback-Leibler divergence (Kullback and Leibler, 1951) between the model and the unknown distribution of the data (see, e.g., Akaike, 1973). Equivalently, it corresponds to maximization of the expected Kullback-Leibler information (or Shannon information or entropy) and thus maximization of the expected information gained (Bernardo, 1979).

An application specific utility may measure the expected benefit or cost, but instead of negating cost (as is usually done), we represent the utilities in a form that is most appealing for the application expert. It should be obvious in each case if a smaller or larger value for the utility is better.

3.1.2 Cross-validation predictive densities

The cross-validation methods for model assessment and comparison have been proposed by several authors: for early accounts see (Stone, 1974; Geisser, 1975) and for more recent review see (Gelfand et al., 1992; Shao, 1993). The cross-validation predictive density dates at least to (Geisser and Eddy, 1979) and nice review of cross-validation and other predictive densities appears in (Gelfand and Dey, 1994; Gelfand, 1996). See also discussion in (Bernardo and Smith, 1994, ch. 6) how cross-validation approximates the formal Bayes procedure of computing the expected utilities.

As the future observations $(x^{(n+h)}, y^{(n+h)})$ are not yet available, we have to approximate the expected utilities by reusing samples we already have. We assume that the future distribution of the data (x, y) is stationary and it can be reasonably well approximated using the (weighted) training data $\{(x^{(i)}, y^{(i)}); i = 1, 2, \dots, n\}$. In the case of conditionally independent observations, to simulate the fact that the future observations are not in the training data, the ith observation $(x^{(i)}, y^{(i)})$ in the training data is left out and then the predictive distribution for $y^{(i)}$ is computed with a model that is fitted to all of the observations except $(x^{(i)}, y^{(i)})$. By repeating this for every point in the training data, we get a collection of leave-one-out cross-validation (LOO-CV) predictive densities

$$\{p(y | x^{(i)}, D^{(\setminus i)}, M); i = 1, 2, \dots, n\}, \tag{3.8}$$

where $D^{(\setminus i)}$ denotes all the elements of D except $(x^{(i)}, y^{(i)})$. To get the LOO-CV-predictive density estimated expected utilities, these predictive densities are compared to the actual $y^{(i)}$'s using utility u, and summarized, for example, with the mean

$$\bar{u}_{\mathrm{LOO}} = E_i[u(y^{(i)}, x^{(i)}, D^{(\setminus i)}, M)]. \tag{3.9}$$

If the future distribution of x is expected to be different from the distribution of the training data, observations could be weighted appropriately (demonstrated in section 3.4.2). By appropriate modifications into the algorithm, the cross-validation predictive densities can also be computed for a data with finite range dependencies (see sections 3.2.3 and 3.4.3).

The LOO-CV-predictive densities are computed with the equation (compare to Equation 3.1)

$$p(y|x^{(i)}, D^{(\setminus i)}, M) = \int p(y|x^{(i)}, \theta, D^{(\setminus i)}, M)p(\theta|D^{(\setminus i)}, M)d\theta. \tag{3.10}$$

For simple models, the LOO-CV-predictive densities may be computed quickly using analytical solutions (see, e.g., Shao, 1993; Orr, 1996; Peruggia, 1997), but models that are more complex usually require a full model fitting for each n predictive densities. When using the Monte Carlo methods it means that we have to sample from $p(\theta|D^{(\setminus i)}, M)$ for each i, and this would normally take n times the time of sampling from the full posterior. If sampling is slow (e.g., when using MCMC methods), the importance sampling LOO-CV (IS-LOO-CV) discussed in section 3.2.1 or the k-fold-CV discussed in section 3.2.3 can be used to reduce the computational burden.

3.1.3 On assumptions made on future data distribution

In this section we briefly discuss assumptions made on future data distribution in the approach described in this work and in related approaches (see, e.g., Rasmussen et al., 1996; Neal, 1998; Dietterich, 1998; Nadeau and Bengio, 2000, and references therein), where the goal is to compare (not assess) the performance of *methods* instead of the *models*.

Assume that the training data D has been produced from the distribution Ω. In model assessment and comparison, we condition the results on given realization of the training data D and assume that the future data for which we want to make predictions comes from the same distribution as the training data, that is, Ω (section 3.1.1).

The method comparison approaches try to answer the question: “Given two methods A and B and training data D, which *method* will produce more accurate *model* when trained on new training data of the same size as D?” (Dietterich, 1998). In probabilistic terms, the predictive distribution of output for every new input in the future is (compare to Equation 3.1)

$$p(y|x^{(n+h)}, D^*_h, M) = \int p(y|x^{(n+h)}, \theta, D^*_h, M) p(\theta|D^*_h, M) d\theta, \qquad (3.11)$$

where D^*_h is the new training data of the same size as D. Although not explicitly stated in the question, all the approaches have assumed that D^*_h can be approximated using the training data D, that is, D^*_h comes from the distribution Ω. The method comparison approaches use various resampling, cross-validation and data splitting methods to produce proxies for D^*_h. The reuse of training samples is more difficult than in the model comparison as the proxies should be as independent as possible in order to be able to estimate well the variability due to a random choice of training data. As the goal of the method comparison is methodological research and not solving a real problem, it is useful to choose problems with large data sets, from which it is possible to select several independent training and test data sets of various sizes (Rasmussen et al., 1996; Neal, 1998). Note that after the method has been chosen and a model has been produced for a real problem, there still is the need to assess the performance of the model.

When solving a real problem, is there a need to retrain the model on new training data of the same size and from the same distribution as D? This kind of situation would rarely appear in practical applications, as it would mean that for every prediction we would use new training data and previously used training data would be discarded. If the new training data comes from the same distribution as the old training data, we could just combine the data and re-estimate the expected utilities. The performance of the model with additional training data could be estimated roughly before getting that data, but it may be difficult because of the difficulties in estimating the shape of the learning curve.

We might want to discard the old training data, if we assume that the future data comes from some other distribution Ω^+ and the new training data D^+ would come from that distribution too. Uncertainty due to using new training data could be estimated in the same way as in method comparison approaches, but in order to estimate how well the results will hold in the new domain we should be able to quantify the difference between the distributions Ω and Ω^+. If we do not assume anything about the distribution Ω^+ we cannot predict the behavior of the model in a new domain as stated by no-free-lunch theorems (see section 2.1.3). Even if the distributions Ω and Ω^+ have just few dimensions, it is very hard to quantify differences and estimate their effect to expected utilities. If the applications are similar (e.g., paper mill and cardboard mill), it may be possible for an expert to give a rough estimate of the model performance in the new domain (it is probably easier to estimate the relative performance of two models than the performance of a single model). In this case, it would be also possible to use information from the old domain as the basis for a prior in the new domain (see, e.g, Spiegelhalter et al., 2000, pp. 18-19 and references therein).

3.2 Methods

In this section, we discuss the importance sampling leave-one-out (section 3.2.1) and k-fold cross-validation (section 3.2.3). We also propose an approach for obtaining samples from the distributions of the expected utilities (section 3.2.4) and discuss model comparison based on expected utilities (section 3.2.5).

3.2.1 Importance sampling leave-one-out cross-validation

In IS-LOO-CV, instead of sampling directly from $p(\theta|D^{(\backslash i)}, M)$, samples $\dot{\theta}_j$ from the full posterior $p(\theta|D, M)$ are reused. Additional computation time in IS-LOO-CV compared to sampling from the full posterior distribution is negligible.

If we want to estimate the expectation of a function $h(\theta)$

$$E(h(\theta)) = \int h(\theta) f(\theta) d\theta, \tag{3.12}$$

and we have samples $\dot{\theta}_j$ from distribution $g(\theta)$, we can write the expectation as

$$E(h(\theta)) = \int \frac{h(\theta) f(\theta)}{g(\theta)} g(\theta) d\theta, \tag{3.13}$$

and approximate it with the Monte Carlo method

$$E(h(\theta)) \approx \frac{\sum_{l=1}^{L} h(\dot{\theta}_j) w(\dot{\theta}_j)}{\sum_{l=1}^{L} w(\dot{\theta}_j)}, \tag{3.14}$$

where the factors $w(\dot{\theta}_j) = f(\dot{\theta}_j)/g(\dot{\theta}_j)$ are called importance ratios or importance weights. See (Geweke, 1989) for the conditions of the convergence of the importance sampling estimates. The quality of the importance sampling estimates depends heavily on the variability of the importance sampling weights, which depends on how similar $f(\theta)$ and $g(\theta)$ are.

A new idea in (Gelfand et al., 1992; Gelfand, 1996) was to use full posterior as the importance sampling density for the leave-one-out posterior densities. By drawing samples $\{\ddot{y}_j; j = 1, \ldots, m\}$ from $p(y|x^{(i)}, D^{(\backslash i)}, M)$, we can calculate the Monte Carlo approximation of the expectation

$$E_y[g(y)|x^{(i)}, D^{(\backslash i)}, M] \approx \frac{1}{m} \sum_{j=1}^{m} g(\ddot{y}_j). \tag{3.15}$$

If $\ddot{\theta}_{ij}$ is a sample from $p(\theta|D^{(\backslash i)}, M)$ and we draw $\ddot{y}_j$ from $p(y|x^{(i)}, \ddot{\theta}_{ij}, M)$, then $\ddot{y}_j$ is a sample from $p(y|x^{(i)}, D^{(\backslash i)}, M)$. If $\dot{\theta}_j$ is a sample from $p(\theta|D, M)$ then samples $\ddot{\theta}_{ij}$ can be obtained by resampling $\dot{\theta}_j$ using importance resampling with weights

$$w_j^{(i)} = \frac{p(\dot{\theta}_j|D^{(\setminus i)}, M)}{p(\dot{\theta}_j|D, M)} \propto \frac{1}{p(y^{(i)}|x^{(i)}, \dot{\theta}_j, D^{(\setminus i)}, M)}. \tag{3.16}$$

In this case, the quality of importance sampling estimates depends on how much the posterior changes when one case is left out.

The reliability of the importance sampling can be estimated by examining the variability of the importance weights. For simple models, the variance of the importance weights may be computed analytically. For example, the necessary and sufficient conditions for the variance of the case-deletion importance sampling weights to be finite for a Bayesian linear model are given by Peruggia (1997). In many cases, analytical solutions are inapplicable, and we have to estimate the efficiency of the importance sampling from the weights obtained. It is customary to examine the distribution of weights with various plots (see Newton and Raftery, 1994; Gelman et al., 1995, ch. 10; Peruggia, 1997). We prefer plotting the cumulative normalized weights (see examples in section 3.4.1). As we get n such plots for IS-LOO-CV, it would be useful to be able to summarize the quality of importance sampling for each i with just one value. For this, we use a heuristic measure of effective sample sizes. Generally, the efficiency of importance sampling depends on the function of interest h (Geweke, 1989). If many different functions h are of potential interest, it is useful to use approximation that does not involve h. The effective sample size estimate based on an approximation of the variance of importance weights can be computed as

$$m_{\text{eff}}^{(i)} = 1 / \sum_{j=1}^{m} (w_j^{(i)})^2, \tag{3.17}$$

where $w_j^{(i)}$ are normalized weights (Kong et al., 1994; Liu and Chen, 1995). We propose to examine the distribution of the effective sample sizes by checking the minimum and some quantiles and by plotting $m_{\text{eff}}^{(i)}$ in increasing order (see examples in section 3.4). Note that this method cannot find out whether the variance of the weights is infinite. However, as the importance sampling is unreliable also with a finite but large variance of weights, the method can be used in practice to estimate the reliability of the importance sampling. In addition, note that a small variance estimate of the obtained sample weights does not guarantee that importance sampling is giving the correct answer, but on the other hand, similar problem applies to any variance or convergence diagnostics method based on finite samples of any indirect Monte Carlo method (see, e.g., Neal, 1993; Robert and Casella, 1999).

Even in simple models like the Bayesian linear model, leaving one very influential data point out may change the posterior so much that the variance of the weights is very large or infinite (Peruggia, 1997). Moreover, even if leave-one-out posteriors are similar to the full posterior, importance sampling in high dimensions suffers from large variation in importance weights (see nice example in (MacKay, 1998b)). Flexible nonlinear models like MLP have usually a high number of parameters and a large number of degrees of freedom (all data points may be influential). We demonstrate in section 3.4.1 a simple case where IS-LOO-CV works well for flexible nonlinear models and in section 3.4.2 a case, which is more difficult and where IS-LOO-CV fails. In section 3.4.3 we illustrate that the importance sampling does not work if data have such dependencies that several samples have to be left out at a time.

In some cases the use of importance link functions (ILF) (MacEachern and Peruggia, 2000) might improve the importance weights substantially. The idea is to use transformations that bring the importance sampling distribution closer to the desired distribution. See (MacEachern and Peruggia, 2000) for an example of computing case-deleted posteriors for Bayesian linear model. For complex models, it may be difficult to find good transformations, but the approach seems to be quite promising.

If there is reason to suspect the reliability of the importance sampling, we suggest using predictive densities from the k-fold-CV, discussed in section 3.2.3.

3.2.2 Importance weights for MLP and GP models

For MLP the predictions are independent of the training data given the parameters of the MLP, so the computing of the importance weights is very straightforward since the term $p(y^{(i)}|x^{(i)}, \dot{\theta}_j, D^{(\setminus i)}, M)$ in Equation 3.16 simplifies to

$$p(y^{(i)}|x^{(i)}, \dot{\theta}_j, D^{(\setminus i)}, M) = p(y^{(i)}|x^{(i)}, \dot{\theta}_j, M). \tag{3.18}$$

For the GP model, the predictions depend on both the parameters of the covariance function and the training data. Thus, the simplification of Equation 3.18 cannot be used. However, the term $p(y^{(i)}|x^{(i)}, \dot{\theta}_j, D^{(\setminus i)}, M)$ in Equation 3.16 can be computed quickly using LOO results for GP with fixed parameters from (Sundararajan and Keerthi, 2001)

$$\log p(y^{(i)}|x^{(i)}, \dot{\theta}_j, D^{(\setminus i)}, M) = \frac{1}{2}\log(2\pi) - \frac{1}{2}\log \bar{c}_{ii} + \frac{1}{2}\frac{q_i^2}{\bar{c}_{ii}}, \tag{3.19}$$

where $\bar{c}_i$ denotes the ith diagonal entry of C^{-1}, q_i denotes the ith element of $q = C^{-1}y$ and C is computed using the parameters $\dot{\theta}_j$. Instead of using the Bayesian approach and integrating over the parameters, Sundararajan and Keerthi (2001) used this result (and its gradient) to find a point estimate for the parameters minimizing the $\sum_{i=1}^{n} \log p(y^{(i)}|x^{(i)}, \dot{\theta}_j, D^{(\setminus i)}, M)$. From results in (Sundararajan and Keerthi, 2001), we also get the mean and the variance of the leave-one-out predictions with given covariance-function parameters

$$E_y[y|x^{(i)}, \dot{\theta}_j, D^{(\setminus i)}, M] = y^{(i)} - \frac{q_i}{\bar{c}_{ii}} \tag{3.20}$$

$$\mathrm{Var}_y[y|x^{(i)}, \dot{\theta}_j, D^{(\setminus i)}, M] = \frac{1}{\bar{c}_{ii}}. \tag{3.21}$$

3.2.3 *k*-fold cross-validation

In k-fold-CV, instead of sampling from n leave-one-out distributions $p(\theta|D^{(\setminus i)}, M)$, we sample only from k (e.g., $k = 10$) k-fold-CV distributions $p(\theta|D^{(\setminus s(i))}, M)$ and then the k-fold-CV predictive densities are computed by the equation (compare to Equations 3.1 and 3.10)

$$p(y|x^{(i)}, D^{(\setminus s(i))}, M) = \int p(y|x^{(i)}, \theta, D^{(\setminus s(i))}, M) p(\theta|D^{(\setminus s(i))}, M) d\theta, \tag{3.22}$$

where $s(i)$ is a set of data points as follows: the data is divided into k groups so that their sizes are as nearly equal as possible and $s(i)$ is the set of data points in group where the ith data point belongs. So approximately n/k data points are left out at a time and thus, if $k \ll n$, computational savings are considerable.

Since the k-fold-CV predictive densities are based on smaller training data sets than the full data set, the expected utility estimate

$$\bar{u}_{\mathrm{CV}} = E_i[u(y^{(i)}, x^{(i)}, D^{(\setminus s(i))}, M)] \tag{3.23}$$

is biased. This bias has been usually ignored, maybe because k-fold-CV has been used mostly in model (method) *comparison*, where biases effectively cancel out if the models (methods) being compared have similar steepness of the learning curves. However, in the case of different steepness of the learning curves and in the model assessment, this bias should not be ignored. To get more accurate results, the bias corrected expected utility estimate $\bar{u}_{\mathrm{CCV}}$ can be computed by using a less well-known first order bias correction (Burman, 1989)

$$\begin{aligned} \bar{u}_{\mathrm{tr}} &= E_i[u(y^{(i)}, x^{(i)}, D, M)] && (3.24) \\ \bar{u}_{\mathrm{cvtr}} &= E_j\big[E_i[u(y^{(i)}, x^{(i)}, D^{(\setminus s_j)}, M)]\big] \quad ; \quad j = 1, \ldots, k && (3.25) \\ \bar{u}_{\mathrm{CCV}} &= \bar{u}_{\mathrm{CV}} + \bar{u}_{\mathrm{tr}} - \bar{u}_{\mathrm{cvtr}}, && (3.26) \end{aligned}$$

where $\bar{u}_{\mathrm{tr}}$ is the expected utility evaluated with the full data given full training data, that is, the training error or the expected utility computed with the posterior predictive densities (see section 3.3.3), and $\bar{u}_{\mathrm{cvtr}}$ is the average of the expected utilities evaluated with the full data given the k-fold-CV training sets. The correction term can be computed by using samples from the full posterior and the k-fold-CV posteriors and no additional sampling is required.

Although the bias can be corrected when k gets smaller, the disadvantage of small k is increased variance of the expected utility estimate. The variance increases with smaller k for the following reasons: the k-fold-CV training data sets are worse proxies for the full training data, there are more ways to divide the training data randomly, but it is divided in just one way, and the variance of the bias correction increases. Values of k between 8 and 16 seem to have good balance between the increased accuracy and increased computational load. In LOO-CV ($k = n$) the bias is usually negligible, but if n is very small it may be useful to compute the bias correction. See discussion in the next section and some related discussion in (Burman, 1989).

We demonstrate in section 3.4.1 a simple case where the IS-LOO-CV and (bias corrected) k-fold-CV give equally good results and in section 3.4.2 a case, which is more difficult where the k-fold-CV works well and the IS-LOO-CV fails. In section 3.4.3, we demonstrate a case where k-fold-CV works but IS-LOO-CV fails, since group dependencies in data require leaving groups of data out at a time.

For the time series with unknown finite range dependencies, the k-fold-CV can be combined with the h-block-CV proposed by Burman et al. (1994). Instead of just leaving the ith point out, additionally a block of h cases from either side of the ith point is removed from the training data for the ith point. The value of h depends on the dependence structure, and it could be estimated for example from autocorrelations. Burman and Nolan (1992) show that $h = 0$ could be used in the case of stationary Markov process and a quadratic form utility (see also Akaike, 1973). However, in real world problems exact properties of the process are not usually known. The approach could also be applied in other models with finite range dependencies (e.g., some spatial models), by removing a block of h cases from *around* the ith point. When more than one data point is left out at a time, importance sampling probably does not work, and either full h-block-CV or k-fold-h-block-CV should be used.

Instead of running full MCMC sampling for each fold in k-fold-CV, it might be possible to reduce the computation time by using coupling of the Markov chains (Pinto and Neal, 2001). In this case, one longer chain would be normally sampled for the full posterior. By coupling the k chains of k-fold-CV to the full posterior chain, shorter chains could be used for the same accuracy.

3.2.4 Estimating the distribution of the expected utility

To assess the reliability of the estimated expected utility, we estimate its distribution. Let us first ignore the variability due to Monte Carlo integration, and consider the variability due to approximation of the future data distribution with a finite number of training data points. We are trying to estimate the expected utilities given the training data D, but the cross-validation predictive densities $p(y|x^{(i)}, D^{(\setminus s_j)}, M)$ are based on training data sets $D^{(\setminus s_j)}$, which are each slightly different. This makes the u_i's slightly dependent in a way that will increase the estimate of the variability of the $\bar{u}$. In the case of LOO-CV, this increase is negligible (unless n is very small) and in the case of k-fold-CV it is practically negligible with reasonable values of k (illustrated in section 3.4.1). If in doubt, this increase could be estimated as mentioned in section 3.4.1. See also comments in the next section.

If utilities u_i are summarized with the mean

$$\bar{u} = E_i[u_i], \tag{3.27}$$

a simple approximation would be to assume u_i's to have an approximately Gaussian distribution (described with the mean and the variance) and to compute the variance of the expected utility of the model as (see, e.g., Breiman et al., 1984, ch. 11)

$$\mathrm{Var}[\bar{u}] = \mathrm{Var}_i[u_i]/n. \tag{3.28}$$

Of course, the distribution of u_i's is not necessarily Gaussian, but still this (or more robust variance estimate based on quantiles) is an adequate approximation in many cases. Variation of this, applicable in the k-fold-CV case, is that first the mean expected utility $\bar{u}_j$ for each k folds is computed and then the variance of the expected utility is computed as (see, e.g., Dietterich, 1998)

$$\mathrm{Var}[\bar{u}] \approx \mathrm{Var}_j[\bar{u}_j]/k. \tag{3.29}$$

Here the distribution of $\bar{u}_j$'s tends to be closer to Gaussian (due to central limit theorem), but the drawback is that this estimator has larger variance than the estimator of Equation 3.28.

If the summary quantity is other than mean (e.g., α-quantile) or the distribution of u_i's is considerably far from Gaussian, above approximations may fail. In addition, the above approximation ignores the uncertainty in the estimates of u_i's due to Monte Carlo error. We propose a quick and generic approach based on Bayesian bootstrap (BB) (Rubin, 1981), which can handle variability due to Monte Carlo integration, bias correction estimation, and the approximation of the future data distribution, as well as arbitrary summary quantities and gives good approximation also in the case of non-Gaussian distributions.

The BB makes a simple non-parametric approximation to the distribution of random variable. Having samples of $z_1, \ldots, z_n$ of a random variable Z, it is assumed that posterior probabilities for the z_i have Dirichlet distribution Di(1,...,1) (see, e.g., Gelman et al., 1995, Appendix A) and values of Z that are not observed have zero posterior probability. Sampling from the Dirichlet distribution gives BB samples from the distribution of the distribution of Z and thus samples of any parameter of this distribution can be obtained. For example, with $\phi = E[Z]$, for each BB sample b we calculate the mean of Z as if $g_{i,b}$ were the probability that

$Z = z_i$; that is, we calculate $\dot{\phi}_b = \sum_{i=1}^{n} g_{i,b} z_i$. The distribution of the values of $\dot{\phi}_b$; $b = 1, \ldots, B$ is the BB distribution of the mean $E[Z]$. See (Lo, 1987; Weng, 1989; Mason and Newton, 1992) for some important properties of the BB.

Assumption that the all possible distinct values of Z have been observed is usually wrong, but with moderate n and not very thick tailed distributions, inferences should not be very sensitive to this unless extreme tail areas are examined. If in doubt, we could use more complex model (e.g., mixture model) that would smooth the probabilities (discarding also the assumption about *a priori* independent probabilities). Of course, fitting parameters of the more complex model would require extra work and it still may be hard to model the tail of the distribution well.

To get samples from the distribution of the expected utility $\bar{u}$, we first sample from the distributions of each u_i (variability due to Monte Carlo integration) and then from the distribution of the $\bar{u}$ (variability due to the approximation of the future data distribution). From obtained samples, it is easy to compute for example credible intervals (CI), highest probability density intervals (HDPI, see Chen et al., 2000), histograms, and kernel density estimates.

Note that the variability due to Monte Carlo integration can be reduced by sampling more Monte Carlo samples, but this can be sometimes computationally too expensive. If the variability due to Monte Carlo integration is negligible, samples from the distributions of each u_i could be replaced by the expectations of u_i.

To simplify computations (and save storage space), we have used thinning to get near independent MCMC samples (estimated by autocorrelations (Neal, 1993, ch. 6; Chen et al., 2000, ch. 3)). However, if MCMC samples were considerably dependent, we could use dependent weights in BB (Künsch, 1989, 1994).

3.2.5 Model comparison with expected utilities

The distributions of the expected utilities can be used for comparing different models. Difference of the expected utilities of two models M_1 and M_2 is

$$\bar{u}_{M_1 - M_2} = E_i[u_{M_1,i} - u_{M_2,i}]. \tag{3.30}$$

If the variability due to Monte Carlo integration is assumed negligible and a Gaussian approximation is used for the distributions of the expected utilities (Equation 3.28 or Equation 3.29), the p-value for the comparison can be computed by using the paired t-test. This approximation was used in case problems in Chapter 2.

With the Bayesian bootstrap, we can sample directly from the distribution of the differences, or if the same random number generator seed has been used for both models when sampling over i (variabilities due to Monte Carlo integrations are independent but variabilities due to the approximations of the future data distribution are dependent through i), we can get samples from the distribution of the difference of the expected utilities as

$$\dot{\bar{u}}_{(M_1-M_2),b} = \dot{\bar{u}}_{M_1,b} - \dot{\bar{u}}_{M_2,b}. \tag{3.31}$$

Then we can, for example, plot the distribution of $\bar{u}_{M_1-M_2}$ or compute the probability $p(\bar{u}_{M_1-M_2} > 0)$ (these probabilities can naturally be combined with prior probabilities of the models). Following simplicity postulate (aka parsimony principle), it is useful to start from simpler models and then test if more complex model would give significantly better predictions. See discussion of simplicity postulate in (Jeffreys, 1961). Note that comparing just point estimates (i.e., assuming that the variance is zero) instead of distributions could easily lead to selection of unnecessarily large models (see, e.g., examples in Chapter 4).

An extra advantage of comparing the expected utilities is that even if there is high probability that one model is better, it might be found out that the difference between the expected utilities still is practically negligible. For example, it is possible that using statistically better model would save negligible amount of money.

Note that a possible overestimation of the variability due to training sets being slightly different (see the previous section) makes these comparisons slightly conservative (i.e., elevated type II error). This is not very harmful, because the error is small and in model choice, it is better to be conservative than too optimistic.

The expected predictive densities have an important relation to Bayes factors, which are commonly used in Bayesian model comparison. If utility u is the predictive log-likelihood and (mean) expected utilities are computed by using cross-validation predictive densities then

$$\mathrm{PsBF}(M_1, M_2) \equiv \prod_{i=1}^{n} \frac{p(y^{(i)}|x^{(i)}, D^{(\setminus i)}, M_1)}{p(y^{(i)}|x^{(i)}, D^{(\setminus i)}, M_2)} = \exp(n\bar{u}_{M_1-M_2}), \tag{3.32}$$

where PsBF stands for pseudo-Bayes factor (Geisser and Eddy, 1979; Gelfand, 1996). As we are interested in the performance of predictions for an unknown number of future samples, we like to report scaled PsBF by taking nth root to get a ratio of "mean" predictive likelihoods (see examples in section 3.4). Note that previously only point estimates for PsBF have been used (except by Vlachos and Gelfand (2000), see below), but with the proposed approach, it is possible to compute also the distribution of the PsBF. The approach can also be used to get samples from other type of Bayes factors, which are briefly discussed in sections 3.3.1 and 3.3.3.

Vlachos and Gelfand (2000) have proposed an approach to estimate the distribution of any model choice criteria $T(D)|M_l$ (*sic*), but the distribution they estimate is not the distribution of the expected PsBF. Vlachos and Gelfand (2000) generate prior predictive replicates $\dot{y}_b^{(i)}; b = 1, \ldots, B$ from the prior predictive distributions $p(y|x^{(i)}, M)$ and then compute, for example, PsBF for each B replicate data set to obtain samples $\dot{t}_b$. We discuss some problems in prior predictive approach in section 3.3.1. See also discussion and example by Gelman et al. (1996) criticizing the use of prior predictive replicates.

As the method we have proposed is based on numerous approximations and assumptions, the results in model comparison should be applied with care when making decisions. However, any selection of a set of models to be compared probably introduces more bias than the selection of one of those models. It should also be remembered that: *"Selecting a single model is always complex procedure involving background knowledge and other factors as the robustness of inferences to alternative models with similar support"* (Spiegelhalter et al., 1998).

3.3 Relations to other approaches

In this section, we discuss the relations of the cross-validation predictive densities to prior predictive densities and Bayes factors (section 3.3.1), other predictive densities (sections 3.3.2 and 3.3.3), and information criteria and the effective number of parameters (section 3.3.4).

3.3.1 Prior predictive densities and Bayes factors

The prior predictive densities (compare to Equation 3.1)

$$p(y|x^{(i)}, M) = \int p(y|x^{(i)}, \theta, M) p(\theta|M) d\theta \tag{3.33}$$

are conditioned only on the prior, not on the data. The expected utilities computed with the prior predictive densities would measure the goodness of the predictions with zero training samples used. Note that in order to have proper predictive densities the prior has to be proper. The expected utilities computed with zero training samples can be used as an estimate of the lower (or upper, if a smaller value is better) limit for the expected utility.

The prior predictive likelihoods

$$\prod_{i=1}^{n} p(y^{(i)}|x^{(i)}, M) = p(D|M) \tag{3.34}$$

are used to compute the Bayes factors (BF)

$$\mathrm{BF}(M_1, M_2) = p(D|M_1)/p(D|M_2), \tag{3.35}$$

which are commonly used in Bayesian model comparison (Jeffreys, 1961; Kass and Raftery, 1995). BF specifically compares the goodness of the priors and

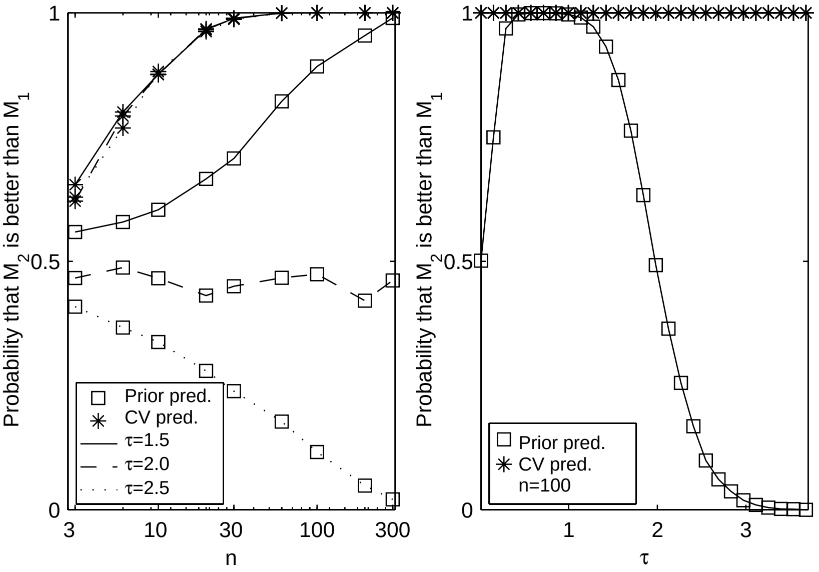

Figure 3.1: The cross-validation predictive vs. the prior predictive (Bayes factor). The true model is $N(1,1)$ and the models compared are $M_1 \sim N(0,1)$ and $M_2 \sim N(\theta,1), \theta \sim N(0,\tau)$. The left plot shows comparison results with different values of τ and with increasing number of data points n. The right plot shows comparison results with $n = 100$ and τ changing. Instead of Bayes factor values, the prior predictive results are also reported as probabilities of the M_2 being better than the M_1.

thus is sensitive to changes in the prior (Jeffreys, 1961; Kass and Raftery, 1995). Therefore, even if the posterior would not be sensitive to changes in the prior, when using BF in model comparison the parameters for the priors have to be chosen with great care.

We illustrate this sensitivity with a small toy problem, where the true model is a normal distribution $N(1,1)$. The first model has no parameters $M_1 \sim N(0,1)$ and thus it cannot get any information from the data. The second model has one parameter $M_2 \sim N(\theta,1), \theta \sim N(0,\tau)$, where θ is the location parameter and τ is the standard deviation of the Gaussian prior on θ. Results were averaged over 1000 realizations of data. The left part of Figure 3.1 shows comparison results with three different values of τ and with increasing number of data points n. The cross-validation predictive approach gives indistinguishable results for all different values of τ, indicating that M_2 is better than M_1, while the prior predictive favors M_1 or M_2 depending on the value of τ. The right part shows comparison results with $n = 100$ and τ changing. The cross-validation predictive approach favors always M_2, while the prior predictive favors M_1 or M_2 depending on τ. It is not shown in the plot, but the cross-validation predictive approach is unable to make difference between models only when $\tau < 10^{-10}$.

This type of prior sensitivity of the Bayes factor has been long known (Jeffreys, 1961) and is called "Lindley's Paradox" or "Bartlett's paradox" (see a nice review and historical comments in (Hill, 1982)).

The prior predictive approach completely ignores how much information is obtained from the data when the prior is updated to the posterior, and consequently, model comparison using the prior predictive densities may produce strange results. However, because it may be possible to estimate unnormalized prior predictive likelihoods for large number of models faster than the expected predictive likelihoods with cross-validation, the prior predictive approach may be used to aid model selection as discussed in Chapter 4.

If prior and likelihood are very different, normalized prior predictive densities may be very difficult to compute (Kass and Raftery, 1995). Relative unnormalized prior predictive likelihoods can be estimated more easily with various methods (see reviews, e.g., in Ntzoufras, 1999; Han and Carlin, 2001). One of such methods is the reversible jump Markov chain Monte Carlo (Green, 1995), which is reviewed in section 4.2.2.

3.3.2 Posterior predictive densities

Posterior predictive densities are naturally used for new data (Equation 3.1). When used for the training data, the expected utilities computed with the posterior predictive densities would measure the goodness of the predictions as if the future data samples would be exact replicates of the training data samples. This is equal to evaluating the training error, which is well known to underestimate the generalization error of flexible models (see also examples in section 3.4). Comparison of the posterior predictive likelihoods $\prod_{i=1}^{n} p(y^{(i)}|x^{(i)}, D, M) = p(D|D, M)$ leads to the posterior Bayes factor (PoBF) (Aitkin, 1991).

The posterior predictive densities should not generally be used neither for assessing model performance, except as an estimate of the upper (or lower if smaller value is better) limit for the expected utility, nor in model comparison as they favor overfitted models (see also discussion of paper (Aitkin, 1991)). Only if $p_{\text{eff}} \ll n$ (see section 3.3.4) the posterior predictive densities may be useful approximation to cross-validation predictive densities, and thus may be used to save computational resources.

The posterior predictive densities are also useful in *Bayesian posterior analysis* advocated, for example, by Rubin (1984), Gelman and Meng (1996), and Gelman et al. (1995, 1996, 2000). In the Bayesian posterior analysis, the goal is to compare posterior predictive replications to the data and examine the aspects of the data that might not accurately be described by the model. Thus, the Bayesian posterior analysis is complementary to the use of the expected utilities in model assessment. Bayesian posterior analysis naturally suffers partly from the same problems as posterior predictive densities generally, that is, using the data twice. However, although posterior predictive densities are overfitted to the training data there still may be many aspects of the data that are not well described and can be detected (see also discussions in references mentioned above). To avoid using the data twice, we have also used the cross-validation predictive densities for such analysis. This approach has also been used by Gelfand et al. (1992); Gelfand (1996), and Draper (1995b, 1996).

The posterior predictive densities also have a connection to information criteria and estimation of the effective number of parameters discussed in section 3.3.4.

3.3.3 Other predictive densities

The partial predictive densities are based on the old idea of dividing the data to two parts, that is, the training and the test set. Comparison of the partial predictive likelihoods $\prod_{i \in s} p(y^{(i)}|x^{(i)}, D^{(\setminus s)}, M) = p(D^{(s)}|D^{(\setminus s)}, M)$ leads to the partial Bayes factor (PaBF) (O'Hagan, 1995). The expected utilities computed with partial predictive densities would correspond to computing only one fold in k-fold-CV, which obviously leads to inferior accuracy.

The fractional Bayes factor (FBF) (O'Hagan, 1995), derived from the partial Bayes factor, is based on comparing fractional marginal likelihoods $q_b(D|M) = \int p^{1-b}(D|\theta, M)\pi_b(\theta|M)d\theta$, where $0 < b < 1$, $p^{1-b}(D|\theta, M)$ is a fractional likelihood and $\pi_b(\theta|M) = p^b(D|\theta, M)p(\theta|M)$ is a fractional posterior (Gilks, 1995). The use of fractions makes it difficult to interpret the FBF in terms of normal predictive densities and expected utilities.

The intrinsic Bayes factor (Berger and Pericchi, 1996) is computed by taking the arithmetic (AIBF) or geometric (GIBF) average or median (MIBF) of all such partial Bayes factors which are computed by using all permutations of minimal subsets of training data that will make the density $p(D^{(s)}|D^{(\setminus s)}, M)$ proper. With a proper prior (which is recommended anyway), intrinsic Bayes factor is the same as (prior) Bayes factor and so the same arguments apply.

3.3.4 Information criteria and the effective number of parameters

Akaike (1969, 1970) proposed to estimate the predictive performance of autoregressive model with final prediction error (FPE), which measures the expected mean square prediction error. Instead of using the full predictive distribution (Equation 3.1), Akaike used the maximum likelihood estimate. Assuming that the stochastic process under consideration is an autoregressive process generated from a strictly stationary and mutually independent innovations, asymptotic second order approximation (Akaike, 1970) gives

$$\mathrm{FPE} = s^2 \left(\frac{1 + p/n}{1 - p/n} \right) \approx s^2 \left(1 + \frac{2p}{n} \right) \tag{3.36}$$

where s^2 is the sample mean square of the residual and p is the number of parameters in the model.

When considering non-predictive models (especially factor analysis, Findley and Parzen, 1995), Akaike (1973) came up with an idea of an information criterion (AIC aka Akaike's information criterion), which is a generalization of the FPE using the log-likelihood (or the deviance which is -2 times the log-likelihood) as the utility. Akaike showed that maximizing the expected log-likelihood corresponds to minimizing the information theoretic Kullback-Leibler divergence between the model and the unknown distribution of the data. Using second order Taylor approximations Akaike (1973) derived asymptotic approximation for the expected log-likelihood given by

$$\mathrm{AIC} = L(\hat{\theta}) - p, \tag{3.37}$$

where $\hat{\theta}$ is the maximum likelihood estimate of θ.

It is also possible to get asymptotic approximations for expected prior and posterior predictive log-likelihoods. Bayesian information criterion (BIC aka Schwarz criterion (Schwarz, 1978)) can be derived from the prior predictive likelihood by using a Laplace approximation for $p(D|M_i)$, neglecting the prior (assuming a very diffuse prior), and taking asymptotic expectations (see, e.g., Gelfand and Dey, 1994) giving

$$\mathrm{BIC} = L(\hat{\theta}) - (\frac{1}{2} \log n) p. \tag{3.38}$$

Similarly, the "posterior information criterion" can be derived from the posterior predictive likelihoods as (Gelfand and Dey, 1994)

$$L(\hat{\theta}) - \frac{1}{2} \log 2. \tag{3.39}$$

As these criteria are based on prior and posterior predictive densities, the problems mentioned in sections 3.3.1 and 3.3.2 apply and thus these criteria are not discussed further.

In both the FPE and the AIC, it is assumed that model is true, $\hat{\theta}$ is a unique solution, and that p does not go to infinity as $n \to \infty$. If the model is wrong (e.g., model is too simple) the approximation is not valid, but it is assumed that in model comparison this error does not favor simpler model over correct model.

Bias correction to the AIC for finite sample has been discussed for example by Hurvich and Tsai (1989, 1991) and Burnham and Anderson (1998). Generalizations of the AIC for *unfaithful* models (i.e., there is no true model) have been discussed for example by Akaike (1981), Chow (1981), Kitagawa (1987), Konishi and Kitagawa (1996), and Burnham and Anderson (1998).

The network information criterion (NIC) by Murata et al. (1994) generalizes the AIC for unfaithful models, arbitrary differentiable utility functions, arbitrary residual models, and penalized or regularized likelihoods (i.e., maximum a posteriori approach in Bayesian terms). Using the second order Taylor approximations, Murata et al. (1994) derive an asymptotic approximation for the expected utility (summarized with mean) given by

$$\bar{u}_{\mathrm{NIC}} = \bar{u}_{\tilde{\theta}} + \mathrm{tr}(K J^{-1}), \tag{3.40}$$

where tr denotes trace, $\tilde{\theta}$ is the maximum a posteriori estimate of θ, $K = \mathrm{Var}[\bar{u}'_{\tilde{\theta}}]$, and $J = \mathrm{E}[\bar{u}''_{\tilde{\theta}}]$. The $\bar{u}'_{\tilde{\theta}}$ and $\bar{u}''_{\tilde{\theta}}$ represent the first and second derivatives with respect to θ. If the model is faithful and the utility is the log-likelihood, then $K = J$, $\mathrm{tr}(K J^{-1}) = p$ and the NIC is the same as the AIC.

FPE, AIC, NIC, and similar criteria approximate the expected utility asymptotically, which will not necessarily give good approximation in a finite case, where the second order approximation may fail. Murata et al. (1994) argue that in a finite case due to ignoring some terms in approximation, these criteria are only effective for model comparison among a sequence of nested models where one is included in another as a lower-dimensional submodel. However, Kitagawa (1987) notes that although the variability of the estimates might be larger, there are no conceptual difficulties in comparing non-nested models.

For linear regression with Gaussian noise assumption, quadratic regularizers, and availability of unbiased estimator, subspace information criterion (SIC) by Sugiyama and Ogawa (2001) gives an unbiased estimate of the squared error with finite samples. SIC has many restrictions but may be useful with small samples in such special cases. Sugiyama and Ogawa (2000) also showed that NIC can be used as an approximation of the SIC.

Murata et al. (1994) note that in the case of model with additive noise NIC reduces to Moody's criteria (Moody, 1992). If the model is unfaithful, $\mathrm{tr}(K J^{-1}) < p$ and Moody (1992) called then $\mathrm{tr}(K J^{-1}) = p_{\mathrm{eff}}$ the *effective* number of parameters. Using the estimate of p_{eff} the expected log-likelihood is given by (compare to Equation 3.37)

$$L(\hat{\theta}) - p_{\mathrm{eff}}. \tag{3.41}$$

In Bayesian models, degrees of freedom in the parameters (and thus p_{eff}) are reduced by the amount of the prior influence and by the amount of dependence between the parameters. p_{eff} depends also on the number of the training samples ($p_{\mathrm{eff}} \leq n$), distribution of the noise in the samples and the complexity of the underlying phenomenon to be modeled. The prior influence and the dependence between the parameters are often substantial and for example in MLPs, it is possible to have $p > n$ but $p_{\mathrm{eff}} \ll n$.

Spiegelhalter et al. (1998, 2001) proposed deviance information criterion (DIC), which is Bayesian generalization of AIC and NIC using the Monte Carlo samples from the posterior for the estimation. Spiegelhalter et al. (2001) justify the DIC heuristically using decision-theoretic arguments, second order Taylor approximations, and comparisons to analytic approximations in certain models. Assuming approximate likelihood normality and using second order Taylor approximation for the deviance and taking the expectations with respect to the posterior distribution of θ gives

$$E_\theta[D(\theta)] \approx D(E_\theta[\theta]) + \mathrm{tr}(-L''V), \tag{3.42}$$

where $V = E_\theta[(\theta - \bar{\theta})(\theta - \bar{\theta})^T]$ is the posterior covariance matrix of θ, and $-L''$ is the observed Fisher's information evaluated at the posterior mean of θ. Under asymptotic posterior normality $E_\theta[\theta] = \tilde{\theta}$, $V = J^{-1}KJ^{-1}$, $-L'' = J$ with deviance as utility, and thus

$$\mathrm{tr}(-L''V) = \mathrm{tr}(KJ^{-1}) = p_{\mathrm{eff}} \tag{3.43}$$

and

$$p_{\mathrm{eff}} \approx E_\theta[D(\theta)] - D(E_\theta[\theta]). \tag{3.44}$$

Note that the inverse of $-L''$ is the likelihood covariance matrix and thus p_{eff} can be thought of as a measure of the ratio of the information in the likelihood about the parameters as a fraction of the total information in the likelihood and the prior (i.e., posterior) (Spiegelhalter et al., 2001).

Spiegelhalter et al. (2001) used Monte Carlo samples from the posterior distribution of θ to estimate $E_\theta[D(\theta)]$ and $D(E[\theta])$ and compute

$$p_{\mathrm{eff,DIC}} = E_\theta[D(\theta)] - D(E_\theta[\theta]) \tag{3.45}$$

$$\mathrm{DIC} = D(E_\theta[\theta]) + 2p_{\mathrm{eff,DIC}}. \tag{3.46}$$

Spiegelhalter et al. (2001) discussed solely the estimation of the expected deviance for model comparison. As discussed in this work, in practical applications it is also useful to use other utilities both in model comparison and especially in model assessment. Spiegelhalter et al. (2001) noted the connection to the NIC incidentally, but did not comment that Monte Carlo samples could be used to estimate the expected utilities with any desired utility. A difficult part in the NIC is the computation of the first and second derivatives in K and J, especially if there is large number of parameters. Using Monte Carlo samples to estimate $E_\theta[\bar{u}(\theta)]$ and $\bar{u}(E_\theta[\theta])$ it is easy to compute an expected utility estimate as

$$\bar{u}_{\mathrm{DIC}} = \bar{u}(E_\theta[\theta]) + 2\left(E_\theta[\bar{u}(\theta)] - \bar{u}(E_\theta[\theta])\right), \tag{3.47}$$

which is a generalization of the DIC in Equation 3.46.

A problem in the DIC approach is that $u(\bar{\theta})$ is not invariant to parameterization (e.g., whether to base $u(\bar{\theta})$ on the posterior means of standard deviations, variances, precisions, log-precisions, or some other choice) and thus the answer depends on over which parameters the expectation is taken. This is problematic, as it is not clear which parameterization should be used and in some cases the effective number of parameters may even be estimated to be negative (Spiegelhalter et al., 2001). Spiegelhalter et al. (2001) recommend improving the likelihood normality in hierarchical models by *focusing*, that is, by taking the expectations over particular set of parameters in the model, which can essentially be used to reduce the model to non-hierarchical structure. In addition, Spiegelhalter et al. (2001) recommend to test different parameterizations and also median parameters beside the mean parameters, which, however, makes the approach harder to analyse.

Note that all the information criteria mentioned above use a *plug-in* predictive distribution (maximum likelihood, maximum a posteriori or posterior mean) rather than the full predictive distribution obtained by integrating out the unknown parameters (Equation 3.1). The *plug-in* predictive distributions ignore the uncertainty about parameter values and model. Spiegelhalter et al. (2001) postulate that in general the use of a plug-in estimate appears to 'cost' an extra penalty of p_{eff}. Asymptotically there is no difference as the uncertainty in parameters vanishes, but as we try to estimate the goodness of the future predictions having only a finite training data set, it is natural to use the full predictive densities.

Stone (1977) considered asymptotic behavior of cross-validation with maximum-likelihood plug-in estimate (which however is asymptotically approximated by the posterior mean). Using a first order Taylor approximation Stone first heuristically showed that the LOO-CV is asymptotically equivalent with the NIC (Stone, 1977, equation 4.5). Stone did not comment usefulness of this intermediate result, but continued showing that it is asymptotically equivalent with the AIC if the model is true and unique solution. As the relation is asymptotic, it does not tell how these methods compare in more complex problems with a finite data set.

In the cross-validation approach, the estimate of the p_{eff} is not needed for model assessment or comparison, but it may provide additional insights into models. We estimate the effective number of parameters by the difference of the expected posterior predictive likelihood

$$\bar{L}_{\mathrm{tr}} = \bar{L}_{\mathrm{po}} = \sum_{i=1}^{n} \log p(y^{(i)}|x^{(i)}, D, M) \tag{3.48}$$

and the expected predictive likelihood estimated with the leave-one-out cross-validation (or with the k-fold-CV)

$$\bar{L}_{\mathrm{LOO}} = \sum_{i=1}^{n} \log p(y^{(i)}|x^{(i)}, D^{(\backslash i)}, M). \tag{3.49}$$

The effective number of parameters can then be estimated in certain cases (see comments below) as

$$p_{\mathrm{eff,LOO}} = \bar{L}_{\mathrm{tr}} - \bar{L}_{\mathrm{LOO}}. \tag{3.50}$$

If $\bar{L}_{\mathrm{LOO}}$ is rewritten as

$$\bar{L}_{\mathrm{LOO}} = \bar{L}_{\mathrm{tr}} - p_{\mathrm{eff,LOO}}, \tag{3.51}$$

the relation to information criteria is obvious.

The approach discussed in section 3.2.4 can also be used to estimate the distribution of $p_{\mathrm{eff,LOO}}$, while the estimation of the distributions of the $p_{\mathrm{eff,DIC}}$ and DIC is still under research (Zhu and Carlin, 2000; Spiegelhalter et al., 2001). There is an asymptotic probability distribution for the NIC (Poncet, 1996), but its accuracy in the finite case is not known. Usually information criteria are used to give only a point estimate of the expected utility. If the point estimate (e.g., mean) of the expected utility would be used for model selection, it would be probable that unnecessarily large models would be selected. Selection of unnecessarily large models is a well-known problem, for example, in the AIC (see, e.g., Shibata, 1976). The difference between the comparison of point estimates and the distributions can be seen in examples in Chapter 4.

This said, DIC might be a useful approximation in model analysis. It has been used also by the author for the models in (Vehtari and Lampinen, 1999b; Järvenpää, 2001). It is faster than the k-fold-CV and might in some cases be more stable than the IS-LOO-CV, but this requires further investigation.

For illustrative purposes, we have reported the effective number of parameters $p_{\mathrm{eff,CV}}$ (with the bias corrected k-fold-CV) and $p_{\mathrm{eff,DIC}}$ (with mean parameterization) for the models used in the illustrative examples in section 3.4.

In MLP networks, there are usually very many correlating parameters, and thus p may be near or even greater than n and $p_{\mathrm{eff}} \ll p$. For MLP models, the use of the DIC approach is straightforward, but for GP models, it is more complicated. Many Gaussian processes can be considered to have an infinite number of parameters, over which the integration is partly done analytically (Neal, 1997, 1999). This makes Gaussian processes very flexible models, and even if the covariance function has only a few parameters, it is possible that the total p_{eff} is larger than the total number of parameters in the covariance function (and the residual model). The cross-validation approach directly produces the total p_{eff} of a Gaussian process, but the direct application of the DIC approach produces the effective number of parameters in the covariance function. This value might also be interesting to know, but with this estimate, the DIC approach would overestimate the expected predictive log-likelihood. The total p_{eff} of a Gaussian process can be approximated in the DIC approach by using samples from latent variables. See (Neal, 1997, 1999) for how to sample latent values of GP. This approach was used in the examples in section 3.4.

In section 3.4.3, we demonstrate a case with group dependencies in the data, which have to be taken into account when estimating the future data distribution. The DIC approach and a cross-validation approach that ignores these dependencies both give a similar, too low estimate for the expected predictive log-likelihood. A cross-validation approach that takes into account these dependencies gives the correct answer for the expected predictive log-likelihood, but then it is possible that $\bar{u}_{\mathrm{tr}} - \bar{u}_{\mathrm{LOO}} > p$.

3.4 Illustrative examples

As illustrative examples, we use MLP networks and Gaussian processes in one toy problem: MacKay's robot arm, and two real world problems: concrete quality estimation and forest scene classification. See (Vehtari and Lampinen, 2001b, Appendix) for details of the models, priors and MCMC parameters.

3.4.1 Toy problem: MacKay's robot arm

In this section we illustrate some basic issues of the expected utilities computed by using the cross-validation predictive densities. A very simple "robot arm" toy-problem (first used by MacKay, 1992) was selected, so that the complexity of the problem would not hide the main points that we want to illustrate. Additionally, we want to demonstrate uncertainties in this problem since this data has been used in many papers without reporting uncertainty in error estimates. Furthermore, it seems probable that different sets of test data have been used in some papers, which has led to overconfident conclusions.

The task is to learn the mapping from joint angles to position for an imaginary robot arm. Two real input variables, x_1 and x_2, represent the joint angles and two real target values, y_1 and y_2, represent the resulting arm position in rectangular coordinates. The relationship between inputs and targets is

$$y_1 = 2.0\cos(x_1) + 1.3\cos(x_1 + x_2) + e_1 \tag{3.52}$$

$$y_2 = 2.0\sin(x_1) + 1.3\sin(x_1 + x_2) + e_2, \tag{3.53}$$

where e_1 and e_2 are independent Gaussian noise variables of standard deviation 0.05. As training data sets, we used the same data sets that were used by MacKay (1992) [1]. There are three data sets each containing 200 input-target pairs which were randomly generated by picking x_1 uniformly from the ranges [-1.932,-0.453] and [+0.453,+1.932], and x_2 uniformly from the range [0.534,3.142]. To get more accurate estimates of the true future utility, we generated additional 10000 input-target pairs having the same distribution for x_1 and x_2 as above, but without noise added to y_1 and y_2. The true future utilities were then estimated using this test data set and integrating analytically over the noise in y_1 and y_2.

We used 8-hidden-unit MLP with 47 parameters and GP model with 4 parameters. In both cases, we used Normal (N) residual model.

[1] Available from http://www.inference.phy.cam.ac.uk/mackay/Bayes_FAQ.html

Figure 3.2 shows the expected utilities where the utility is root mean square error. The IS-LOO-CV and the 10-fold-CV give quite similar error estimates. Figure 3.3 shows that the importance sampling works probably very well for the GP but it might produce unreliable results for the MLP. Although importance sampling weights for the MLP are not very good, the IS-LOO-CV results are not much different from the 10-fold-CV results in this simple problem. Note that in this case, small location errors and even a large underestimation of the variance in the IS-LOO-CV predictive densities are swamped by the uncertainty from not knowing the noise variance.

In Figure 3.2 also the realized, estimated, and theoretical noise in each data set is shown. Note that the estimated error is lower if the realized noise is lower and the uncertainty in estimated errors is about the same size as the uncertainty in the noise estimates. This demonstrates that most of the uncertainty in the estimate of the expected utility comes from not knowing the true noise variance. Figure 3.4 verifies this, as it shows the different components that contribute to the uncertainty in the estimate of the expected utility. The variability due to having slightly different training sets in the 10-fold-CV and the variability due to the Monte Carlo approximation are negligible compared to the variability due to not knowing the true noise variance. The estimate of the variability due to having slightly different training sets in the 10-fold-CV was computed by using the knowledge of the true function. In real world cases where the true function is unknown, this variability could be approximated using the CV terms calculated for the bias correction, although this estimate might be slightly optimistic. The estimate of the variability due to Monte Carlo approximation was computed directly from the Monte Carlo samples using the Bayesian bootstrap. Figure 3.4 also shows that bias in the 10-fold-CV is quite small. As the true function was known, we also computed estimates for the biases using the test data. For all the GPs, the bias corrections and the "true" biases were the same with about 2% accuracy. For the MLPs, there was much more variation, but still the "true" biases were inside the 90% credible interval of the bias correction estimate. Although in this example there would be no practical difference in reporting the expected utility estimates without the bias correction, bias may be significant in other problems. For example, in the examples of sections 3.4.2 and 3.4.3 the bias correction had practically notable effect.

Figure 3.5 demonstrates the difficulty of estimating the extrapolation capability of the model. As the distribution of the future data is estimated with the training data, it is not possible to know how well the model would predict outside the training data. If it is possible to affect the data collection, it is advisable to make sure that enough data is collected from the borders of assumed future data distribution, so that extrapolation for future predictions could be avoided.

Figures 3.6 and 3.7 demonstrate the comparison of models using paired comparison of the distributions of the expected utilities. Figure 3.6 shows the ex-

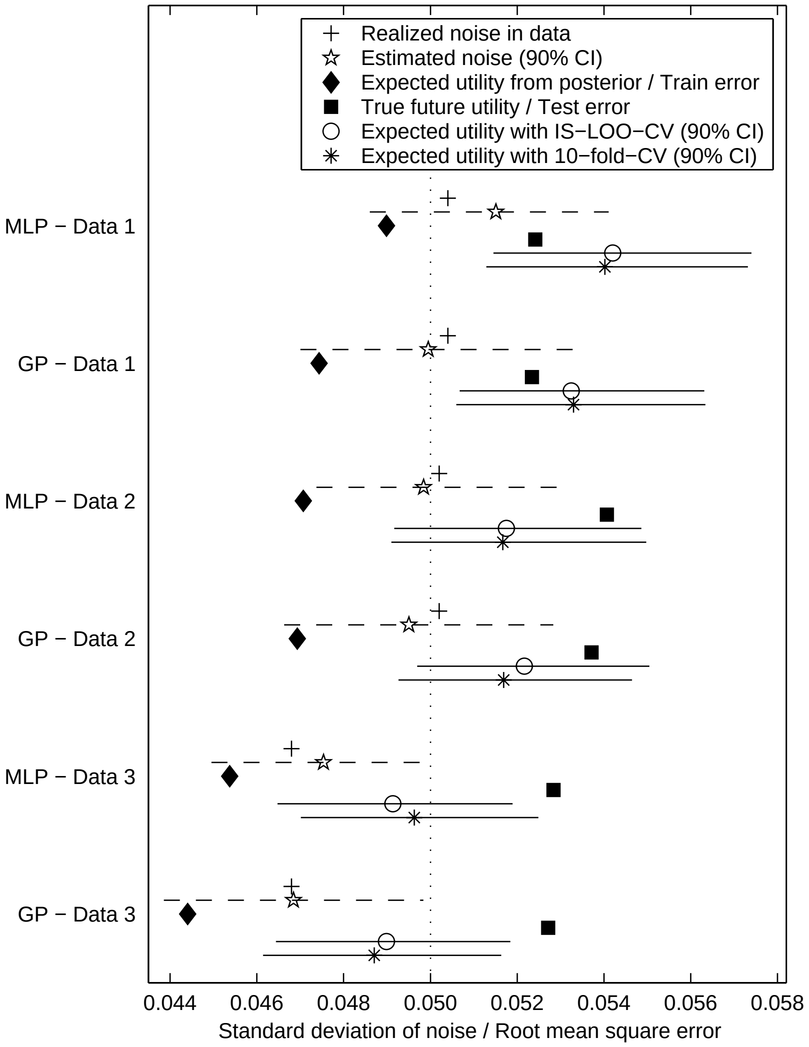

Figure 3.2: Robot arm example: The expected utilities (root mean square errors) for MLPs and GPs. Results are shown for three different realizations of the data. The IS-LOO-CV and the 10-fold-CV give quite similar error estimates. Realized noise and estimated noise in each data set is also shown. Dotted vertical line shows the level of the theoretical noise. Note that the estimated error is lower if the realized noise is lower and the uncertainty in estimated errors is about the same size as the uncertainty in the noise estimates.

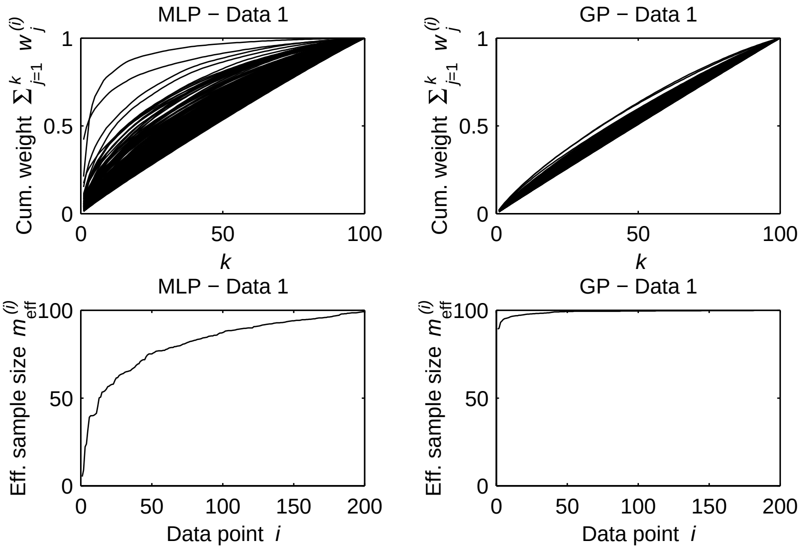

Figure 3.3: Robot arm example: Two plot types were used to visualize the reliability of the importance sampling. Top plots show the total cumulative mass assigned to the k largest importance weights versus k (one line for each data point i). The MLP has more mass attached to fewer weights. Bottom plots show the effective sample size of the importance sampling m^i_{eff} for each data point i (sorted in increasing order). The MLP has less effective samples. These two plots show that in this problem, the IS-LOO-CV may be unreliable for the MLP, but probably works well for the GP.

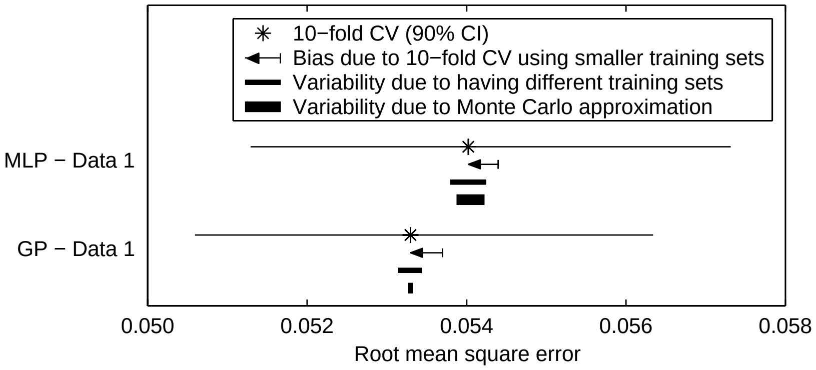

Figure 3.4: Robot arm example: The different components that contribute to the uncertainty, and bias correction for the expected utility (root mean square errors) for MLP and GP. Results are shown for the data set 1. The variability due to having slightly different training sets in 10-fold-CV and the variability due to the Monte Carlo approximation are negligible compared to the variability due to not knowing the true noise variance. The bias correction is quite small, as it is about 0.6% of the mean error and about 6% of the 90% credible interval of error.

pected difference of root mean square errors and Figure 3.7 shows the expected ratio of mean predictive likelihoods (nth root of the pseudo-Bayes factors). The IS-LOO-CV and the 10-fold-CV give quite similar estimates, but disagreement shows slightly more clearly here when comparing models than when estimating expected utilities (compare to Figure 3.2). The disagreement between the IS-LOO-CV and the 10-fold-CV might be caused by bad importance weights of the IS-LOO-CV for the MLPs (see Figure 3.3).

Figure 3.8 shows different components that contribute to the uncertainty in paired comparison of the distributions of the expected utilities. The variability due to having slightly different training sets in the 10-fold-CV and the variability due to the Monte Carlo approximation have larger effect in pairwise comparison, but they are almost negligible compared to the variability due to not knowing the true noise variance. Figure 3.8 also shows that in this case, the bias in the 10-fold-CV is negligible.

Figure 3.9 shows the effective number of parameters for the MLP and GP models used in this problem. The estimates were computed with the k-fold-CV and the DIC with mean parameterization. In this case the latent variable approximation used to get the total $p_{\mathrm{DIC,eff}}$ of GP fails. The effective number of parameters in the covariance function of the GP models estimated with the DIC (see section 3.3.4) was about 2. This is quite sensible, as there was only one residual model parameter and the three covariance function parameters correlate strongly.

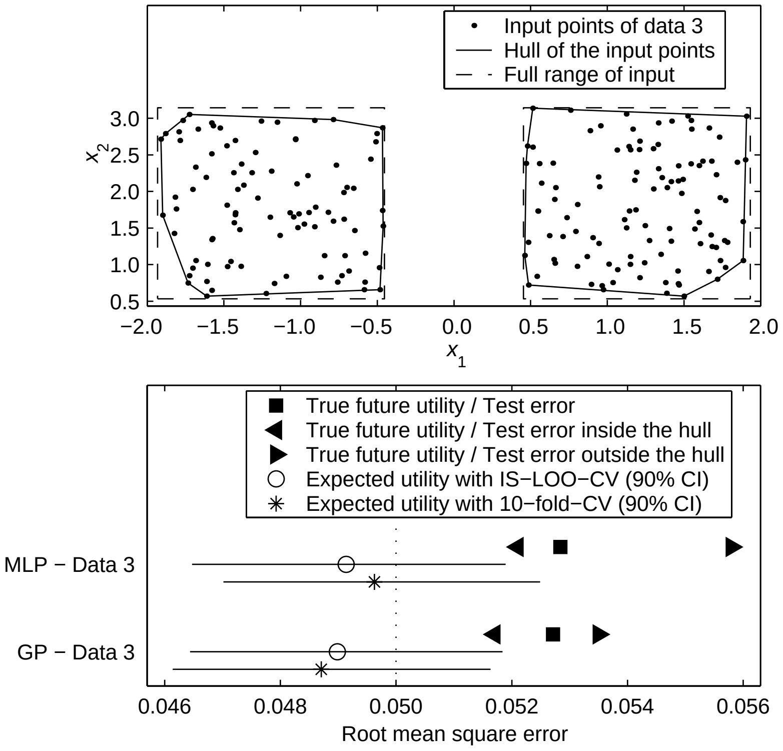

Figure 3.5: Robot arm example: The upper plot shows input points of data set 3, with the full range (broken line) and with the realized range approximated by two convex hulls (solid line). The lower plot shows how the true future utility (test error) inside the hull coincides better with credible interval for estimated expected utility.

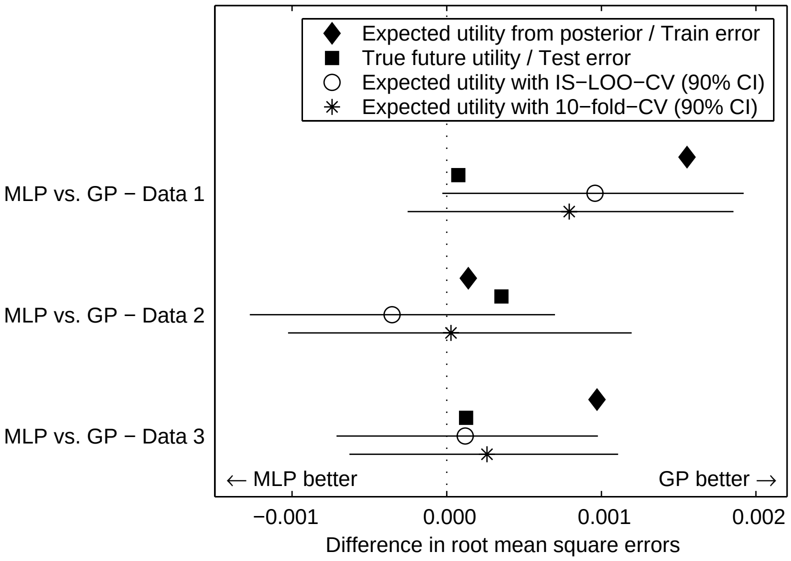

Figure 3.6: Robot arm example: The expected difference of root mean square errors for MLP vs. GP. Results are shown for three different realizations of the data. The disagreement between the IS-LOO-CV and the 10-fold-CV shows slightly more clearly when comparing models than when estimating expected utilities (compare to Figure 3.2). Figure 3.3 indicates reason to suspect the reliability of the IS-LOO-CV.

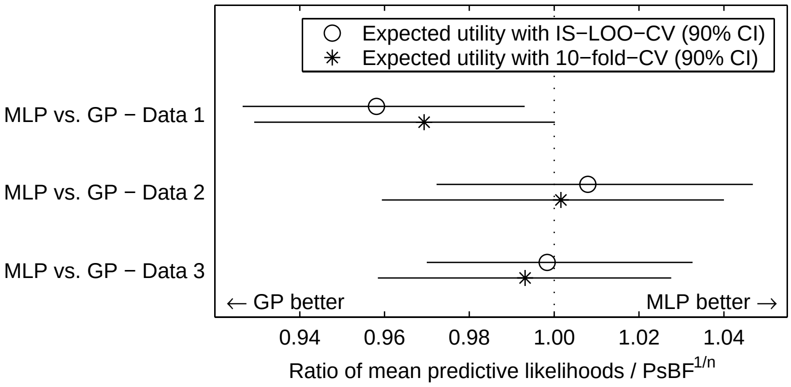

Figure 3.7: Robot arm example: The expected ratio of mean predictive likelihoods (nth root of the pseudo-Bayes factors) for MLP vs. GP. Results are shown for three different realizations of the data. The disagreement between the IS-LOO-CV and the 10-fold-CV shows slightly more clearly when comparing models than when estimating expected utilities (compare to Figure 3.2). Figure 3.3 indicates reason to suspect the reliability of the IS-LOO-CV.

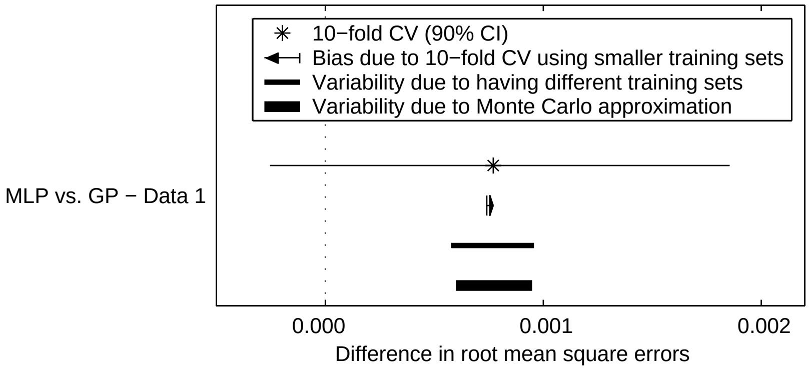

Figure 3.8: Robot arm example: The different components that contribute to the uncertainty, and bias correction for the expected difference of the expected root mean square errors for MLP vs. GP. Results are shown for the data set 1. The variability due to having slightly different training sets in the 10-fold-CV and the variability due to the Monte Carlo approximation are almost negligible compared to the variability from not knowing the true noise variance. In this case, the biases effectively cancel out and the combined bias correction is negligible.

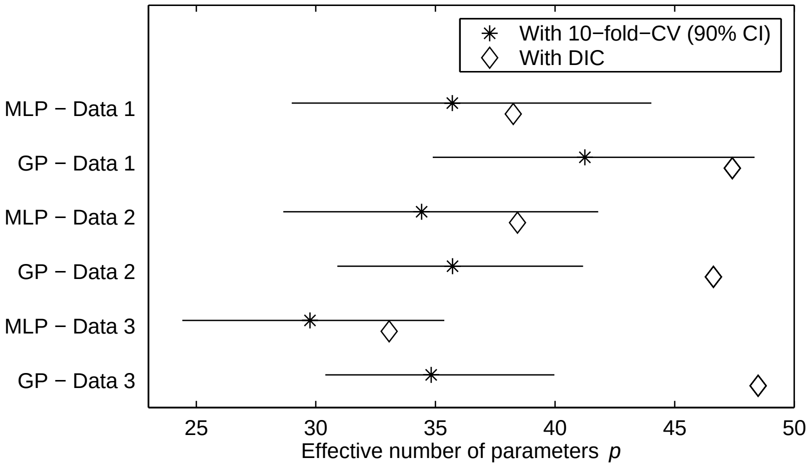

Figure 3.9: Robot arm example: The effective number of parameters for the MLP and GP models estimated with the *k*-fold-CV and the DIC (with mean parameterization).

3.4.2 Case I: Concrete quality estimation

In this section we present results from a real world problem of predicting the quality properties of concrete (described in section 2.3.1). In the following we report the results for the volume percentage of air in the concrete, air-%. Similar results were obtained for the other variables.

We tested 10-hidden-unit MLP networks and GP models with Normal (N), Student's t_ν, input dependent Normal (in.dep.-N) and input dependent t_ν residual models. The Normal model was used as standard reference model and Student's t_ν, with an unknown degrees of freedom ν, was used as longer tailed robust residual model that allows a small portion of samples to have large errors. When analyzing results from these two first residual models, it was noticed that the size of the residual variance varied considerably depending on three inputs, which were zero/one variables indicating the use of additives. In the input dependent residual models, the parameters of the Normal or Student's t_ν were made dependent on these three inputs with common hyperprior.

Figure 3.10 shows the expected normalized root mean square errors and the expected 90%-quantiles of absolute errors for MLP and GP with Normal (N) residual model. The root mean square error was selected as general discrepancy utility and the 90%-quantile of absolute error was chosen after discussion with the concrete expert, who preferred this utility as it is easily understandable. The IS-LOO-CV gives much lower estimates for the MLP and somewhat lower estimates for the GP than the 10-fold-CV. Figure 3.11 shows that the IS-LOO-CV for both MLP and GP has many data points with small (or very small) effective sample size, which indicates that the IS-LOO-CV cannot be used in this problem.

Figure 3.12 shows the expected normalized root mean square errors, the expected 90%-quantiles of absolute errors and the expected mean predictive likelihoods for GP models with Normal (N), Student's t_ν, input dependent Normal (in.dep.-N) and input dependent t_ν residual models. There is not much difference in expected utilities if root mean square error is used (it is easy to guess the mean of prediction), but there are larger differences if mean predictive likelihood is used instead (it is harder to guess the whole predictive distribution). The bias corrections are not shown but they were about 3-5% of the median values, that is, they have notable effect in model assessment. The biases were similar in different models, so they more or less effectively cancel out in model comparison.

Tables 3.1, 3.2, and 3.3 show the results for the pairwise comparisons of the residual models. In this case, the uncertainties in the comparison of the normalized root mean square errors and the 90%-quantiles of absolute errors are so big that no clear difference can be made between the models. As we get similar performance with all models (measured with these utilities), we could choose anyone of them without the fear of choosing a bad model. With the mean predictive likelihood utility, there is more difference as it also measures the goodness

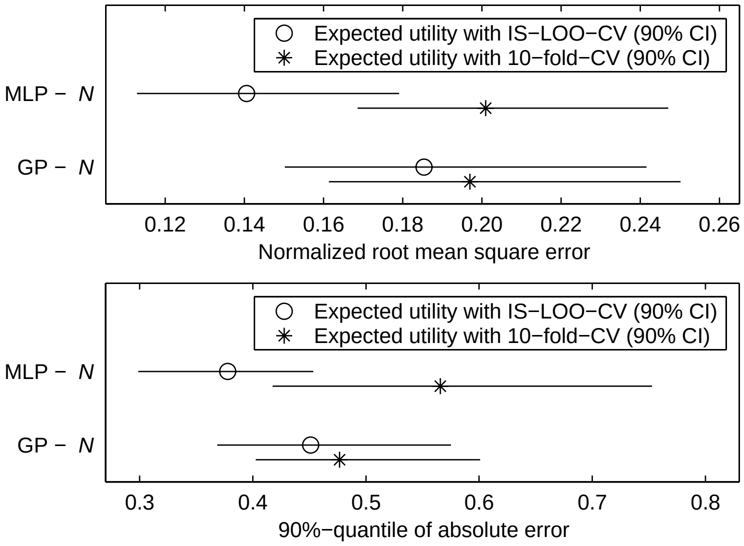

Figure 3.10: Concrete quality estimation example: The expected utilities for MLP and GP with the Normal (N) residual model. The top plot shows the expected normalized root mean square errors and the bottom plot shows the expected 90%-quantiles of absolute errors. The IS-LOO-CV gives much lower estimates for the MLP and somewhat lower estimates for the GP than the 10-fold-CV. Figure 3.11 indicates reason to distrust the IS-LOO-CV.

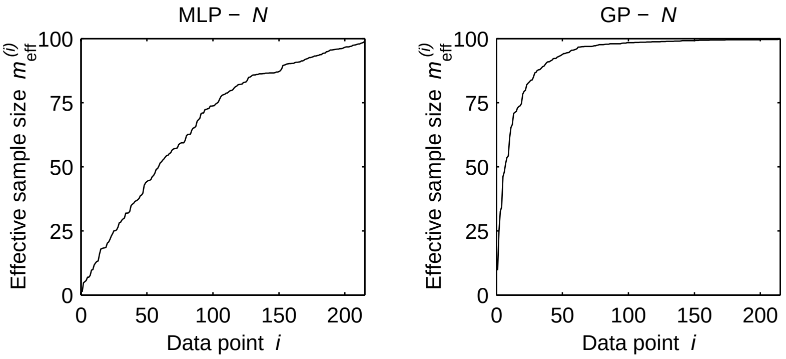

Figure 3.11: Concrete quality estimation example: The effective sample sizes of the importance sampling $m_{\mathrm{eff}}^{(i)}$ for each data point i (sorted in increasing order) for MLP and GP with the Normal (N) noise model. Both models have many data points with a small effective sample size, which implies that the IS-LOO-CV cannot be trusted.

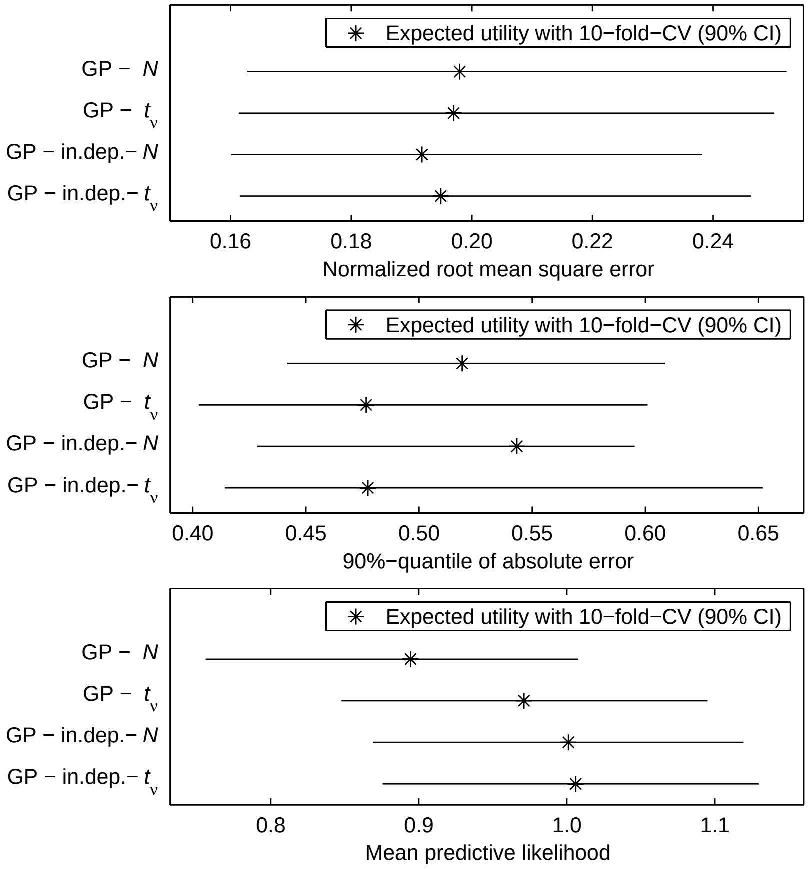

Figure 3.12: Concrete quality estimation example: The expected utilities for GP models with Normal (N), Student's t_ν, input dependent Normal (in.dep.-N) and input dependent t_ν residual models. The top plot shows the expected normalized root mean square errors (smaller value is better), the middle plot shows the expected 90%-quantiles of absolute errors (smaller value is better) and the bottom plot shows the expected mean predictive likelihoods (larger value is better). There is not much difference in expected utilities of different residual models if root mean square error is used as utility (it is easy to guess the mean of the prediction), but there are larger differences if mean predictive likelihood is used instead (it is harder to guess the whole predictive distribution). See Tables 3.1, 3.2, and 3.3 for the pairwise comparisons of the residual models.

Table 3.1: Concrete quality estimation example: Pairwise comparison of GP models with different residual models using the normalized root mean square error as utility (see also Figure 3.12). The values in the matrix are probabilities that the model in the row is better than the model in the column. Uncertainties in the predictive utilities are so big (see also Figure 3.12) that no clear difference can be made between the residual models using the normalized root mean square error as utility.

		Comparison			
residual model		1.	2.	3.	4.
1.	N		0.40	0.22	0.33
2.	t_ν	0.60		0.18	0.31
3.	input dependent N	0.78	0.82		0.85
4.	input dependent t_ν	0.67	0.69	0.15	

Table 3.2: Concrete quality estimation example: Pairwise comparison of GP models with different residual models using the 90%-quantile of absolute error as utility (see also Figure 3.12). The values in the matrix are probabilities that the model in the row is better than the model in the column. Uncertainties in the predictive utilities are so big (see also Figure 3.12) that no clear difference can be made between residual models using the 90%-quantile of absolute error as utility.

		Comparison			
residual model		1.	2.	3.	4.
1.	N		0.17	0.53	0.21
2.	t_ν	0.83		0.87	0.67
3.	input dependent N	0.47	0.13		0.23
4.	input dependent t_ν	0.79	0.33	0.77	

Table 3.3: Concrete quality estimation example: Pairwise comparison of GP models with different residual models using mean predictive likelihood as utility (see also Figure 3.12). The values in the matrix are probabilities that the model in the row is better than the model in the column. It seems quite probable that the input dependent t_ν residual model is better than N or t_ν and is not much better than input dependent N.

		Comparison			
Residual model		1.	2.	3.	4.
1.	N		0.02	0.01	0.00
2.	t_ν	0.98		0.22	0.06
3.	input dependent N	0.99	0.78		0.32
4.	input dependent t_ν	1.00	0.94	0.68	

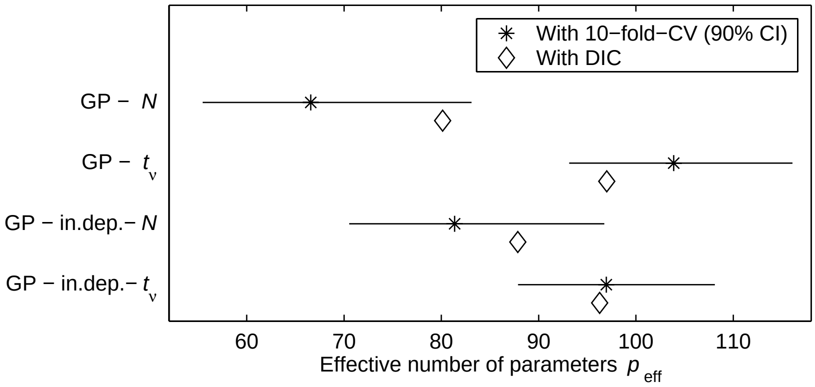

Figure 3.13: Concrete quality estimation example: The effective number of parameters for the GP models with Normal (N), Student's t_ν, input dependent Normal (in.dep.-N) and input dependent t_ν residual models. The plot shows the estimates for the effective number of parameters computed with the k-fold-CV and the DIC.

of the tails. If in addition to point estimates, the predictive distributions (or, e.g., credible intervals for predictions) are wanted, input dependent t_ν model would be probably the best choice.

Figure 3.13 shows the effective number of parameters for the GP models used above. There were 30–258 model parameters depending on the residual model in those models. In this case, the approximation used in the DIC to get the total $p_{\mathrm{DIC,eff}}$ of GP seemed to work reasonably. The effective number of parameters in the covariance function of the GP models estimated with the DIC were about 6, 39, 8 and 27, respectively. For comparison, in the MLP models there were 322–335 model parameters (note that $p > n$) depending on the residual model and the effective numbers of parameters were estimated to be about 75–90.

Knowing that the additives have strong influence on the quality of concrete it was useful to report also the expected utilities separately for samples with different additives (i.e. assuming that in all future casts no additives or just one of the additives will be used). Figure 3.14 shows for the GP with input dependent t_ν residual model the expected 90%-quantiles of absolute errors for samples with no additives, with additive A or B, and all samples. Samples with the additive B have much larger expected error. This is natural as the additive B is used to increase the air-% in concrete and the variation in the measurements is relative to the value of the air-%.

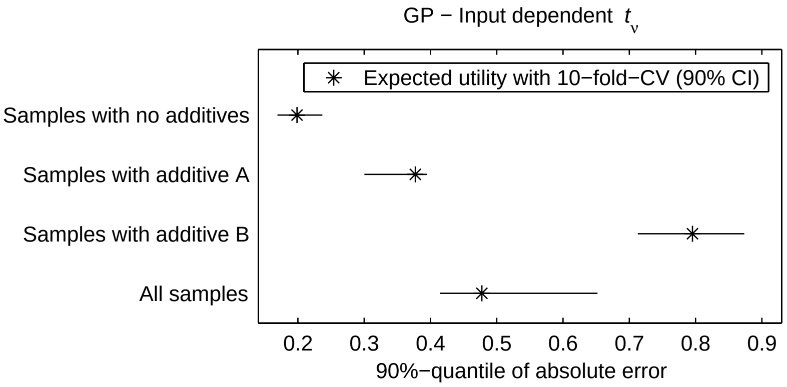

Figure 3.14: Concrete quality estimation example: The expected utilities for the GP with the input dependent t_ν residual model. The plot shows the expected 90%-quantiles of absolute errors for samples with no additives, with additive A or B, and all samples. As the additives have effect on the amount of variation in the quality of concrete, it is natural that value of the expected utility also depends on additives.

3.4.3 Case II: Forest scene classification

In this section, we illustrate that if, due to dependencies in the data, several data points should be left out at a time, *k*-fold-CV has to be used to get more accurate results. The case problem is the classification of forest scenes with an MLP network described in section 2.3.2.

Textures and lighting conditions are more similar in different parts of one image than in different images. If the LOO-CV is used or data points are divided randomly in the *k*-fold-CV, training and test sets may have data points from the same image, which would lead to over-optimistic estimates of the predictive utility. This is caused by the fact that instead of having 4800 independent data points, we have 48 sample images which each have 100 highly dependent data points. This increases our uncertainty about the future data. To get a more accurate estimate of the predictive utility for new unseen images, training data set has to be divided by images.

We tested two 20-hidden-unit MLPs with logistic likelihood model. The first MLP used all 84 inputs and the second MLP used a reduced set of 18 inputs selected using the reversible jump MCMC (RJMCMC) method (see Chapter 4).

As discussed in section 3.2.1 and demonstrated in section 3.4.2, leaving one point out can change posterior so much that importance sampling does not work. Leaving one image (100 data points) out will change posterior even more. Figure 3.15 shows the effective sample sizes of the importance sampling for the 84-input MLP for the IS-LOO-CV and the IS-LOIO-CV (leave-one-image-out). For the 18-input MLP the result was similar. In this case, neither the IS-LOO-CV nor the IS-LOIO-CV can be used.

The expected classification errors for the 84 and 18-input MLPs are shown in Figure 3.16. The predictive utilities computed by using the posterior predictive densities (training error) give too low estimates. The IS-LOO-CV and the 8-fold-CV with random data division give too low estimates because the data points from one image are highly dependent. The IS-LOO-CV also suffers from somewhat bad importance weights and the IS-LOIO-CV suffers from very bad importance weights (see Figure 3.15). In the group 8-fold-CV, the data division was made by handling all the data points from one image as one indivisible group. The bias corrections are not shown but they were for the 84 and 18 input MLPs about 9% and 3% of the median values, respectively. Note that the more complex model had naturally a steeper learning curve and correspondingly a larger bias correction. In this case, biases did not cancel out in model comparison.

The pairwise comparison computed by using the group 8-fold-CV predictive densities gave a probability of 0.86 that the 84-input model has lower expected classification error than the 18-input model. We still might use the smaller model for classification, as it would be not much worse, but slightly faster.

Figure 3.17 shows the expected mean predictive likelihoods and the effec-

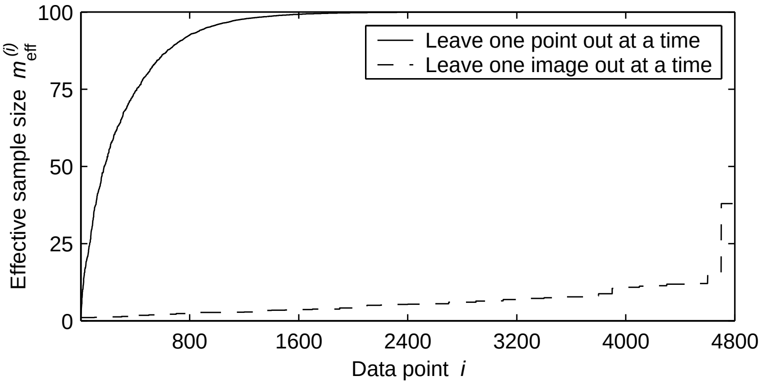

Figure 3.15: Forest scene classification example: The effective sample sizes of the importance sampling $m_{\mathrm{eff}}^{(i)}$ for each data point i (sorted in increasing order) for the 84-input logistic MLP. The effective sample sizes are calculated both for the leave-one-point-out (IS-LOO-CV) and the leave-one-image-out (IS-LOIO-CV) methods. In both cases there are many data points with a small effective sample size, which implies that importance sampling cannot be trusted in this problem.

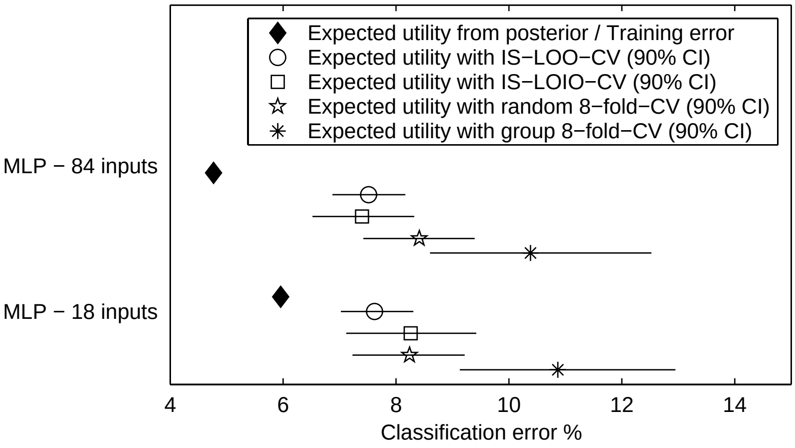

Figure 3.16: Forest scene classification example: The expected utilities (classification errors) for the 84 and 18-input logistic MLPs. The IS-LOO-CV gives too low estimates because the data points from one image are highly dependent (and also because of bad importance weights). The IS-LOIO-CV gives too low estimates because of bad importance weights when leaving one image out at time (see Figure 3.15). The 8-fold-CV with random data division gives too low estimates because the data points from one image are highly dependent. The group 8-fold-CV gives better estimates, as the data division was made by handling all the data points from one image as one indivisible group.

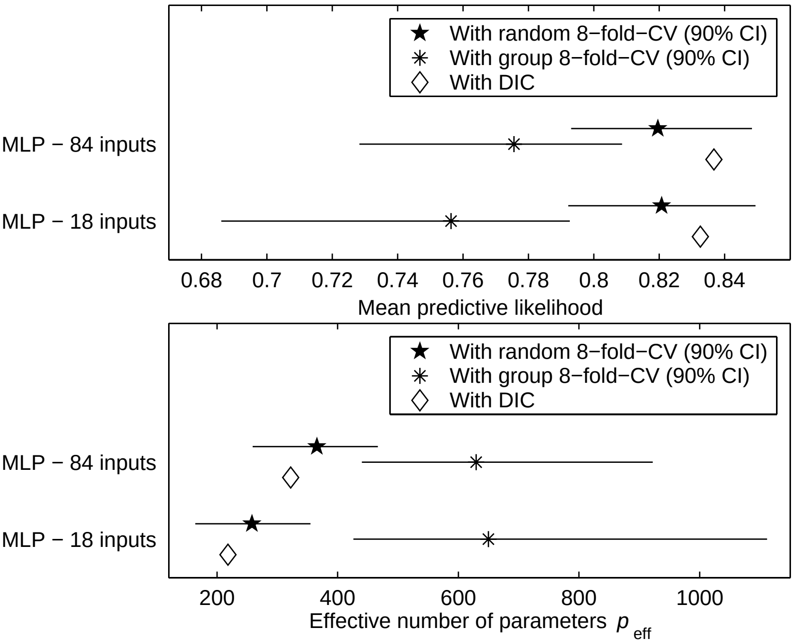

Figure 3.17: Forest scene classification example: The top plot shows the expected mean predictive likelihoods computed with the 8-fold-CV and the DIC. The bottom plot shows the estimates for the effective number of parameters computed with the k-fold-CV and the DIC. The 84-input MLP had $p = 1721$ and the 18-input MLP had $p = 401$.

tive number of parameters for the MLP models used above. There were 1721 and 401 parameters for 84 and 18-input MLPs, respectively. The 8-fold-CV with random data division and the DIC give too optimistic estimates of the expected mean predictive likelihoods because data points from one image are highly dependent. The group 8-fold-CV gives good estimates for the expected mean predictive likelihoods, but now the difference between the posterior predictive likelihood and the expected predictive likelihood is not the same as the effective number of parameters. This can be seen clearly in the 18-input MLP, for which $p(\bar{u}_{\text{tr}} - \bar{u}_{\text{LOO}} > p) = 0.98$.

Chapter 4

Input variable selection using expected utilities

4.1 Introduction

In practical problems, it is often possible to measure many variables, but it is not necessarily known which of them are relevant and required to solve the problem. In Bayesian models, it is usually feasible to use large number of potentially relevant input variables by using suitable priors with hyperparameters controlling the effect of the inputs in the model (see Chapter 2). Although such models may have good predictive performance, it may be difficult to analyse them, or costly to make measurements or computations. To make the model more explainable (easier to gain scientific insights) or to reduce the measurement cost or the computation time, it may be useful to select a smaller set of input variables. In addition, if the assumptions of the model and prior do not match well the properties of the data, reducing the number of input variables may even improve the performance of the model. Our goal is to find a model with the smallest number of input variables having statistically or practically at least the same predictive ability as the full model with all the available inputs.

With just a few models, it is easy to compare them using the cross-validation approach. In the case of K inputs, there are 2^K input combinations, and computing the cross-validation predictive densities for each model easily becomes computationally prohibitive. For example, in our second example case (section 4.3.2), with 84 inputs, even if a million models could be checked in a second, going through all the combinations would take over 600 million years. As the computation of the cross-validation predictive densities may take minutes or hours for each model candidate, we need some way to reduce the number of potential models that are compared to just a few models.

We propose to use variable dimension jump Markov chain Monte Carlo methods to find out potentially useful input combinations, for which the final model choice and assessment is done using the cross-validation predictive densities. We have used the reversible jump Markov chain Monte Carlo (RJMCMC) method (Green, 1995), which is one of the simplest to implement and one of the fastest on big problems. The RJMCMC visits models according to their posterior probabilities, and thus models with negligible probability are probably not visited in finite time. Consequently, only the most probable models are investigated and computational savings can be considerable compared to going through all possible models.

The posterior probabilities of the models, given by the RJMCMC, are proportional to the product of the prior probabilities of the models and the prior predictive likelihoods of the models. The prior predictive likelihood measures the goodness of the model if no training data were used and thus can be used to estimate the lower limit of the expected predictive likelihood. In model comparison, the predictive likelihood is a useful utility, as it measures how well the model predicts the predictive distribution. This way it is possible to choose a smaller set of models in a reasonable amount of time for which better estimates of the expected utilities (with any desired utility) can be computed using the cross-validation predictive densities. We review the RJMCMC method in section 4.2.2.

Estimates based on the prior predictive densities computed in many ways (including the RJMCMC), have been used directly for model selection (including the input variable selection) (see, e.g., Kass and Raftery, 1995; Ntzoufras, 1999; Sykacek, 2000; Han and Carlin, 2001; Kohn et al., 2001; Chipman et al., 2001, and the references therein). In section 3.3.1 we discussed the sensitivity of the prior predictive densities to the choice of the priors of the model parameters. In section 4.2.1, we discuss why it is not possible to be uninformative when choosing priors for the model space and how information about the prior probabilities of the number of input variables can be included using various priors. Furthermore, Spiegelhalter (1995) argues that when selecting single model from some family of models instead of integrating over the discrete model choices (e.g., input combinations), it is better to compare the consequences (e.g., utilities) of the models instead of their posterior probabilities. Consequently, we argue that the posterior probabilities of the models should not usually be used directly for input selection.

If there are many correlated inputs, it is probable that there are also many high-probability input combinations and thus it may be hard to get enough samples in reasonable time to estimate the probabilities of input combinations well. In this case, we propose to use the marginal probabilities of the inputs, which are easier to estimate, to indicate potentially useful inputs. This is illustrated in section 4.3.2.

In addition to input selection, the marginal probabilities of inputs, given by the RJMCMC, can be used to estimate the relevance of the inputs, which has great importance in analyzing the final model. In section 2.2.4 we discussed and demonstrated the fact that instead of the predictive importance, nonlinearity of the input has the largest effect on the ARD values. In section 4.3.1 we also illustrate the difference between the marginal probabilities of inputs and the ARD values for relevance estimation.

The proposed approach is generic, and faster methods might be used for simple regular models, for example by combining analytic approximations of $p(D|M_l)$ or $\bar{u}_{M_l}$ with stochastic optimization (e.g., simulated annealing) over l. Although not demonstrated here, if extremely parsimony MLP models are desired, the proposed approach could also be used to select the connections and the number of hidden units in MLP network.

As illustrative examples, we use MLP networks and Gaussian processes in two real world problems (section 4.3).

4.2 Methods

In this section we discuss prior issues specific in input selection (section 4.2.1) and review the RJMCMC method, which can be used to obtain estimates of the (unnormalized) prior predictive likelihoods for a huge amount of models in a time comparable to computing the cross-validation predictive densities for a single model (section 4.2.2).

4.2.1 Priors for input selection

In this section we discuss the model space priors for input selection $p(M_l|I)$, where I denotes the assumptions about the model space. We also discuss the effect of parameter priors $p(\theta_{M_l}|M_l)$ to the posterior probabilities of the input combinations.

If we have K potential inputs, there are $L = 2^K$ different models. A simple and popular choice is the uniform prior on models

$$p(M_l) \equiv 1/L, \tag{4.1}$$

which is noninformative in the sense of favoring all models equally, but as seen below, will typically not be noninformative with respect to the model size.

It will be convenient to index each of the 2^K possible input combinations with the vector

$$\gamma = (\gamma_1, \dots, \gamma_K)^T, \tag{4.2}$$

where γ_k is 1 or 0 according to whether the input k is included in the model or not, respectively. We get equal probability for all the input combinations (models) by setting

$$p(\gamma) = (1/2)^K. \tag{4.3}$$

From this we can see that the implicit prior for the number of inputs k is the Binomial

$$p(k) = \mathrm{Bin}(K, 1/2), \tag{4.4}$$

which clearly is not uninformative, as $E[k] = 0.5K$ and $\mathrm{Var}[k] = 0.25K$. For example, if K=27, then k lies in the range 7 to 20 with prior probability close to 1 (see also examples in section 4.3).

To favor smaller models various priors on the number of inputs (or other components) have been used; for example, geometric (Rios Insua and Müller, 1998), truncated Poisson (Phillips and Smith, 1996; Denison et al., 1998; Sykacek, 2000), and truncated Poisson with a vague Gamma hyperprior for λ (Andrieu et al., 2000). A problem with these approaches is that the implicit Binomial prior still is there, producing the combined prior

$$p(k) = \mathrm{Bin}(K, 1/2)h(k), \tag{4.5}$$

where $h(k)$ is the additional prior on the number of inputs. Although it is possible to move the mass of the prior to favor a smaller number of inputs with the additional prior, the Binomial prior effectively restricts k *a priori* to lie in a short range.

Instead of an additional prior on the number of inputs, we could set the probability of single input being in the model, π, to the desired value and get

$$p(\gamma) = \pi^k (1 - \pi)^{1-k} \tag{4.6}$$

and correspondingly

$$p(k) = \mathrm{Bin}(K, \pi). \tag{4.7}$$

In this case, $E(k|\pi) = K\pi$ and $\mathrm{var}(k|\pi) = K\pi(1 - \pi)$. Although having more control, this would still be quite informative about the number of inputs k.

A more flexible approach is to place a hyperprior on π. Following Kohn et al. (2001) and Chipman et al. (2001), we use a Beta prior

$$p(\pi) = \mathrm{Beta}(\alpha, \beta), \tag{4.8}$$

which is convenient, as then the prior for k is Beta-binomial

$$p(k) = \text{Beta-bin}(n, \alpha, \beta). \tag{4.9}$$

In this case, $E[k|\pi, \alpha, \beta] = K\frac{\alpha}{\alpha+\beta}$ and $\mathrm{Var}[k|\pi, \alpha, \beta] = K\frac{\alpha\beta(\alpha+\beta+K)}{(\alpha+\beta)^2(\alpha+\beta+1)}$, and thus the values for α and β are easy to solve after setting $E[k]$ and $\mathrm{Var}[k]$ to the desired values. As the Beta-binomial is often nonsymmetric, it may be easier to choose the values for α and β by plotting the distribution with different values of α and β, as we did in the examples in section 4.3. If $\alpha = 1$ and $\beta = 1$ then the prior on k is uniformly distributed on $(0, K)$, but now the models are not equally probable, as the models with few or many inputs have higher probability than the models with about $K/2$ inputs. Consequently, it is not possible to be uninformative in input selection, and some care should be taken when choosing priors, as efforts to be uninformative in one respect will force one to be informative in other respect.

Above we have assumed that each input has equal probability. This assumption could be relaxed by using, for example, a prior of the form

$$p(\gamma) = \prod \pi_k^{\gamma_k}(1-\pi_k)^{1-\gamma_k}, \tag{4.10}$$

where π_k is the probability of input k being in the model. This kind of prior could be further combined with a hierarchical prior on π_k to gain more flexibility. It seems that prior information about the relative probabilities of the inputs is rarely available, as this kind of priors are seldom used.

In some cases there might be information about dependencies between input combinations that could be used. For example, dependency priors in the case of related input variables are discussed by Chipman (1996). Although we know that the inputs in our case problems are not independent, we do not know *a priori* what dependencies there might be, so we use the independence prior. Additionally, as one of our goals is to get more easily explainable models, it is desired that inputs that are as independent as possible are selected. In section 4.3, we illustrate some problems arising in input selection when inputs are not independent.

As discussed and illustrated in section 3.3.1, the prior for parameters $p(\theta_{M_l}|M_l)$ greatly affects the posterior probability of the model M_l having extra parameter $\theta^+_{M_l}$. If the prior on the extra parameters $p(\theta^+_{M_l}|M_l)$ is too tight, the extra parameters might not reach a useful range in the posterior, thus making the model less probable. On the other hand, if the prior is too vague, the probability of any value for the extra parameter gets low, and correspondingly, the probability of the model gets low. This kind of prior sensitivity in input selection has been discussed and demonstrated for example by Richardson and Green (1997),Dellaportas and Forster (1999), and Ntzoufras (1999). Often, it is recommended to test different priors, but there is no formal guidance what to do if the different priors produce different results. Some methods for controlling the effects of the prior in linear models are discussed by Ntzoufras (1999), but these methods may be difficult to generalize to other models. Using hierarchical priors seems to alleviate partly the problem, as discussed by Richardson and Green (1997) and illustrated in section 4.3. Furthermore, since the effect of the model space prior is considerable and its selection usually quite arbitrary, there is probably no need to excessively fine tune the priors of the parameters in question. Naturally, the prior sensitivity is an even smaller problem when the final model choice is based on the expected utilities computed by using the cross-validation predictive densities.

4.2.2 Reversible jump Markov chain Monte Carlo

The reversible jump Markov chain Monte Carlo (RJMCMC) (Green, 1995) is an extension to the Metropolis-Hastings method allowing jumps between models with different dimensional parameter spaces. In the case of input selection, models have different number of parameters as they have different number of inputs. When adding or removing inputs, the corresponding parameters are added or removed, respectively.

If the current state of the Markov chain is (M_1, θ_{M_1}) the jump to state (M_2, θ_{M_2}) is accepted with probability

$$\alpha = \min\left(1, \frac{p(D|\theta_{M_2}, M_2)p(\theta_{M_2}|M_2)p(M_2|I)j(M_2, M_1)q(t_2|\theta_{M_2}, M_2, M_1)}{p(D|\theta_{M_1}, M_1)p(\theta_{M_1}|M_1)p(M_1|I)j(M_1, M_2)q(t_1|\theta_{M_1}, M_1, M_2)} \times \left|\frac{\partial h_{M_1,M_2}(\theta_{M_1}, t_1)}{\partial(\theta_{M_1}, t_1)}\right|\right), \tag{4.11}$$

where $j(M_1, M_2)$ is the probability of jumping from M_1 to M_2, q is the proposal distribution for the parameter t and h_{M_1,M_2} is an invertible function defining the mapping between the parameter spaces, $(\theta_{M_2}, t_2) = h_{M_1,M_2}(\theta_{M_1}, t_1)$.

In the case of suitable proposal distribution, the acceptance probability term can be greatly simplified. When adding a new input, we set h_{M_1,M_2} as identity, that is, $\theta_{M_2} = (\theta_{M_1}, t_1)$, and use the conditional prior of the new parameters as the proposal distribution. Then the Jacobian determinant is 1, the prior terms for the parameters common to both models cancel out, and the prior and the proposal distribution for the new parameters cancel out. Moreover, as we set $j(M_1, M_2) = j(M_2, M_1)$, Equation 4.11 simplifies to

$$\alpha = \min\left(1, \frac{p(D|\theta_{M_2}, M_2)p(M_2|I)}{p(D|\theta_{M_1}, M_1)p(M_1|I)}\right). \tag{4.12}$$

We use hierarchical priors for the parameters specific to inputs, and so the conditional prior is adapting to the data. Thus, the conditional prior is a natural proposal distribution with a reasonable acceptance rate and mixing behavior.

To make convergence diagnostics and estimation of credible intervals easier, it is useful to run several (e.g., 10) independent RJMCMC chains (with different starting points) for each case. For between-model convergence diagnostics, we used in our case problems the chi-squared and Kolmogorov-Smirnov tests proposed by Brooks et al. (2001), which also utilize several independent chains. As there was large number of models, the number of visits to each model was typically very low, and thus we analysed the visits to each subpopulation having equal number of inputs. The Kolmogorov-Smirnov test seemed to be the best in revealing between-model convergence problems. For other convergence assessment methods for the RJMCMC, see (Brooks and Giudici, 1999, 2000).

The RJMCMC is not the only method suitable for estimating the unnormalized prior predictive likelihoods, but it is one of the simplest to implement and one of the fastest on big problems. The RJMCMC visits models according to their posterior probabilities, and models with negligible probability are probably not visited in a finite time. This way only the most probable models are investigated and computational savings can be considerable compared to going through all the possible models. Some alternative methods have been reviewed, for example, by Ntzoufras (1999) and Han and Carlin (2001).

4.3 Illustrative examples

As illustrative examples, we use MLP networks and Gaussian processes in two real world problems: concrete quality estimation (section 4.3.1) and forest scene classification (section 4.3.2). We first describe the additional implementation details.

In the RJMCMC, we had two model change proposals. The first one was the commonly used proposal, which proposes to change state γ_k of the random input k. The second was used to improve mixing. It proposed to simultaneously remove one random input for which $\gamma_k = 1$ and add one random input for which $\gamma_k = 0$. Although not in fact changing the dimensionality, it was handled in a same way as the first proposal.

To improve conditional priors of the input parameters, a previously fixed degree of freedom in the prior for the ARD values was changed to have its own prior distribution (having most of the mass between 0.4 and 4). This change did not have significant effect on the predictive distributions of the single models, but it seemed to slightly improve the proposal distributions for the RJMCMC. In the case of MLPs, the sampling for that extra variable was made by discretized Gibbs and in the case of GPs with the HMC.

We also experimented with the conditional maximization and the auxiliary variable methods (Brooks et al., 2000), but we could not improve the acceptance rates despite of some tuning attempts. Finding the conditional maximum was too slow and unstable while the auxiliary variable method easily got stuck despite tuning attempts.

4.3.1 Case I: Concrete quality estimation

In this section, we present results from the real world problem of predicting the quality properties of concrete (described in section 2.3.1).

The aim of the study was to identify which properties of the stone material are important, and additionally, examine the effects that properties of the stone material have on concrete. It was desirable to get both the estimate of relevance of all available input variables and select a minimal set required to get a model with statistically the same predictive capability as with the full model. A smaller model is easier to analyze and there is no need to make possibly costly or toxic measurements in the future for properties having negligible effect. The problem is complicated because there are strong cross-effects, and the inputs measuring similar properties have strong dependencies.

For models used in (Järvenpää, 2001), we had made the input selection using the DIC (see section 3.3.4) and heuristic backward selection. Although this approach produced reasonable results, it required a full model fitting for each model investigated, contained some *ad hoc* choices to speed up the heuristic backward selection, and lacked clear results for the relevance of the different inputs. Below we present results using the RJMCMC and the expected utilities computed by using the cross-validation predictive densities. With this approach, we were able to get more insight about the problem, smaller models, and improved reliability of the results.

We used 10-hidden-unit MLP networks and GP models. The residual model used was input dependent Student's t_ν with unknown degrees of freedom ν. As the size of the residual variance varied depending on three inputs, which were zero/one variables indicating the use of additives, the parameters of the Student's t_ν were made dependent on these three inputs with a common hyperprior.

We report here detailed results for the *air-%* and less detailed results for the *bleeding*. See (Vehtari and Lampinen, 2001a) for additional results for the *flow value*, and the *compressive strength*, for which similar results were obtained.

In the case of air-% and GP with both uniform and Beta priors, the between-model jump acceptance was about 6% and in the case of MLP with uniform and Beta priors, the between-model jump acceptance was about 5% and 1.5%, respectively. To increase the effective jump acceptance, between-model jumps were made three times more probable than in-model jumps. In the case of GP, from about 10^8 possible input combinations, the 4000 saved states visited about 3500 and 2500 different input combinations with uniform and Beta priors, respectively. Few most probable models were visited by all ten independent chains and for example, ten most probable models were visited by at least eight chains. Thus, useful credible intervals could be computed for the model probabilities.

We first report results for the GP model predicting the air-%. Figures 4.1 and 4.2 show the posterior probabilities of the number of inputs with an equal prior probability for all the models and with Beta-bin(27, 5, 10) prior on the number of inputs, respectively. With equal prior probability for all models, the prior probability for the number of inputs being less than eight is so low that it is unlikely that the RJMCMC will visit such models. Parameters for the Beta-binomial prior were selected to better reflect our prior information, that is, we thought it might be possible to have a low number of inputs, most probably about 6-12 inputs and not excluding the possibility for a larger number of inputs. Note that the

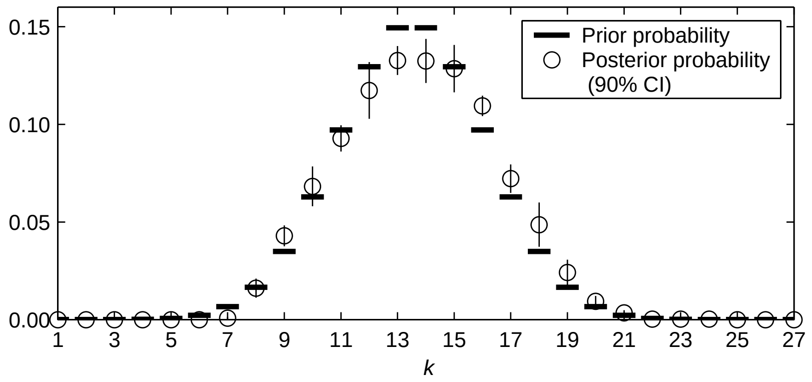

Figure 4.1: Concrete quality estimation example, predicting the air-% with GP: The posterior probabilities of the number of inputs with "uninformative" prior, i.e., equal prior probability for all models.

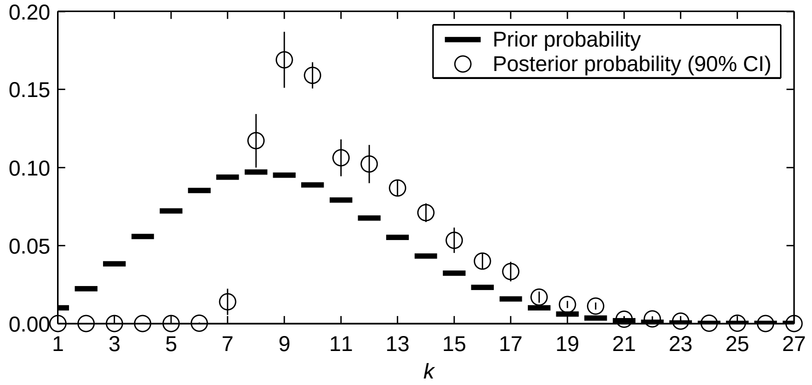

Figure 4.2: Concrete quality estimation example, predicting the air-% with GP: The posterior probabilities of the number of inputs with Beta-bin(27, 5, 10) prior on the number of inputs.

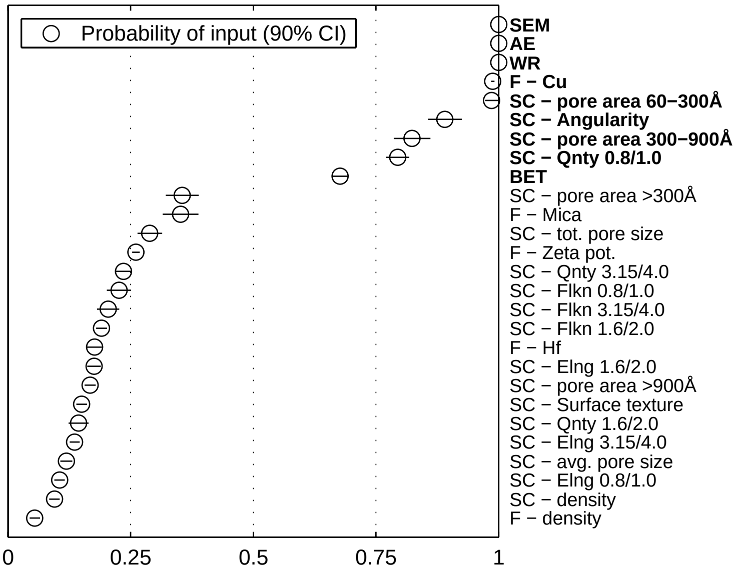

Figure 4.3: Concrete quality estimation example, predicting the air-% with GP: The marginal posterior probabilities of the inputs with a Beta-bin(27,5,10) prior on the number of inputs. The inputs in the most probable model are in boldface (see Figure 4.5).

Beta-binomial prior used is in fact more vague about the number of inputs than the “uninformative” prior. The posterior distribution of the number of inputs is quite widespread, which is natural as the inputs are dependent and the ARD prior allows use of many inputs.

Figure 4.3 shows the marginal posterior probabilities of the inputs with a Beta-bin(27,5,10) prior on the number of inputs. The nine most probable inputs are clearly more probable than the others and the other inputs have posterior probability approximately equal to or less than the mean prior probability of an input (1/3).

Figure 4.4 shows the ARD values of the inputs for the full model. Eight of the nine most probable inputs have also a larger ARD value than the other inputs, but they cannot be clearly distinguished from the other inputs. Moreover, input *“BET”* (measuring the specific surface area of the fines) is ranked much lower by the ARD than by the probability (compare to Figure 4.3). Further investigation revealed that *“BET”* was relevant, but had near linear effect.

Figure 4.5 shows the posterior probabilities of the ten most probable input combinations with a Beta-bin(27,5,10) prior on the number of inputs. All the ten models are very similar, only minor changes are present in few inputs, and, the changed inputs are known to correlate strongly. In this case, two models

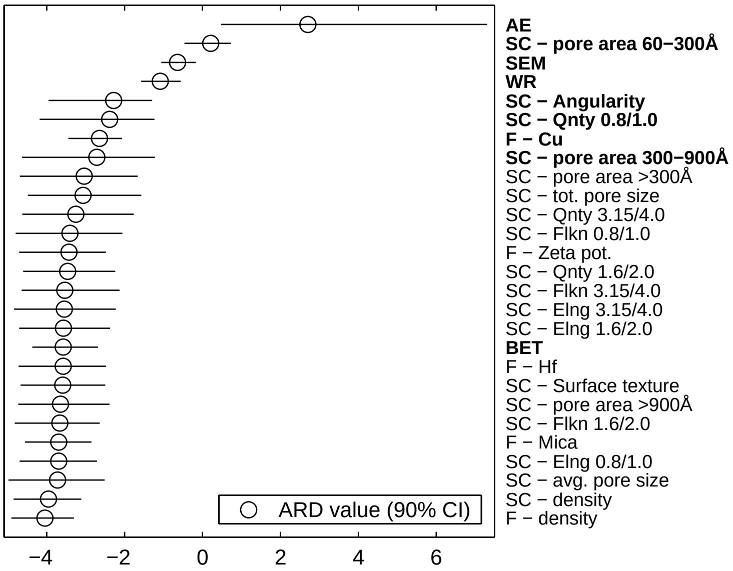

Figure 4.4: Concrete quality estimation example, predicting the air-% with GP: The ARD values of the inputs of the full model. The nine most probable inputs are in boldface. Compare to Figure 4.3.

are significantly more probable than others, but between them, there is no clear difference. As the other probable models are similar to the two most probable models, it is likely that the probability mass has been spread to many equally good models.

For the final model choice, we computed the expected utilities using the cross-validation predictive densities for the most probable models. Differences between the most probable models and the full model were small, and so there was no big danger of choosing a bad model. To verify that by conditioning on single model we do not underestimate the uncertainty about the structure of model (see, e.g., Draper, 1995a; Kass and Raftery, 1995), we also computed the expected utility for the model, in which we integrated over all the possible input combinations. Such integration can readily be approximated using the previously obtained RJMCMC samples. There was no significant difference in the expected likelihoods.

To illustrate the differences between the prior-predictive and the cross-validation predictive densities, Figure 4.6 shows the expected utilities computed using the cross-validation predictive densities for the full model and the models having the k ($k = 5, \ldots, 15$) most probable inputs. Note that the expected predictive likelihoods are similar for models having at least about eight most probable inputs, while prior predictive likelihoods (posterior probabilities) are very different

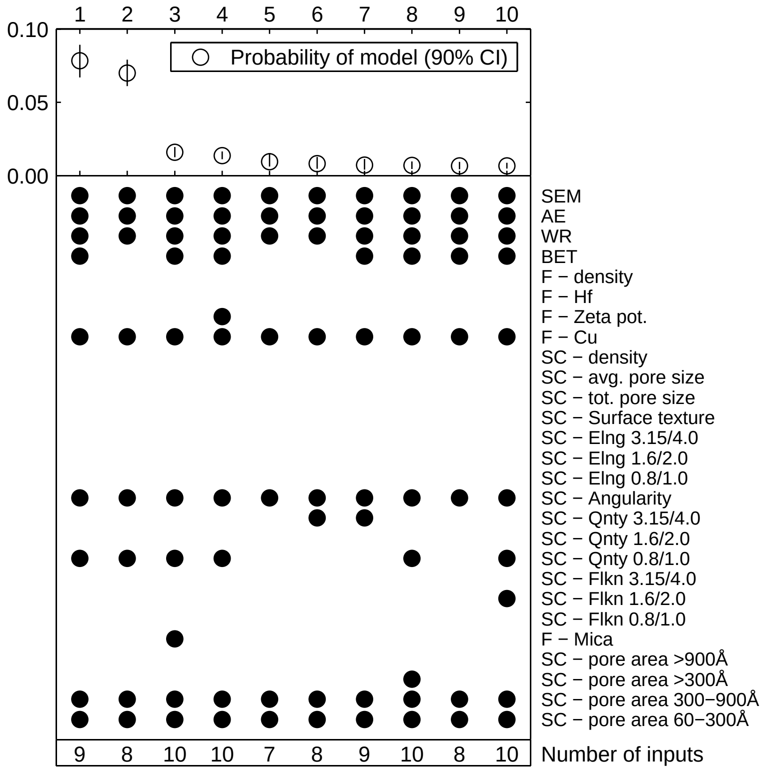

Figure 4.5: Concrete quality estimation example, predicting the air-% with GP: The probabilities of the ten most probable models with a Beta-bin(27,5,10) prior on the number of inputs. The top part shows the probabilities of the models, the middle part shows which inputs are in the model, and the bottom part shows the number of inputs in the model.

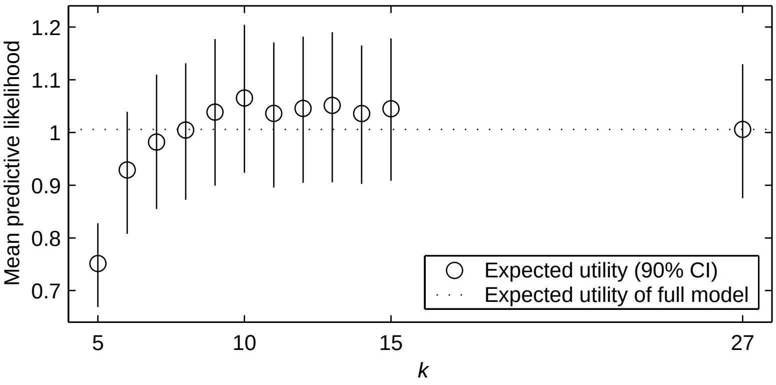

Figure 4.6: Concrete quality estimation example, predicting the air-% with GP: The expected utilities (mean predictive likelihoods) of the models having the *k* most probable inputs (see Figure 4.3). After about nine inputs, adding more inputs does not improve the model performance significantly. To give an impression of the differences in pairwise comparison, there is for example about 90% probability that the nine input model has a higher predictive likelihood than the eight input model.

for models with different number of inputs. For example, the prior predictive likelihood of the full model is vanishingly small compared to the most probable models, but the cross-validation predictive likelihood is similar to the most probable models. The performance of the full model is similar to smaller models, as the ARD type prior allows many inputs without reduced predictive performance. Note that if the point estimate (e.g., mean) of the expected utility would be used for model selection, larger models would be selected than when selecting the smallest model with statistically the same utility as the best model. The problem of selecting larger models than necessary is a common problem when estimating the expected utilities with information criteria that give just a point estimate (see section 3.3.4).

To illustrate the effect of the prior on approximating functions, we also report results for input selection with MLP. The results for the MLP were not sensitive to changes in the hyperparameter values, so the difference in the results is probably caused mainly by the difference in the form of the covariance function realized by the GP and MLP models.

Figure 4.7 shows the posterior probabilities of the number of the inputs with a Beta-bin(27,5,10) prior on the number of inputs. In the case of MLP, larger number of inputs is more probable than in the case of GP (compare to Figure 4.2). Figure 4.8 shows the marginal posterior probabilities of the inputs with a Beta-bin(27,5,10) prior on the number of inputs. Most of the inputs have higher posterior probabilities than the mean prior probability (1/3). There is no clear division

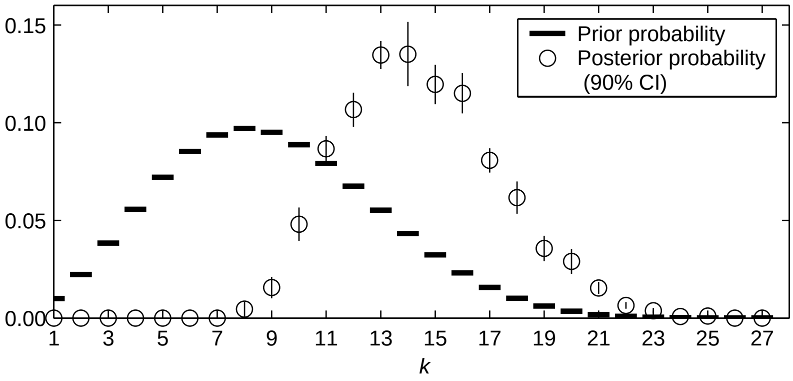

Figure 4.7: Concrete quality estimation example, predicting the air-% with MLP: The posterior probabilities of the number of inputs with a Beta-bin(27,5,10) prior on the number of inputs. Compare to the results for the GP in Figure 4.2.

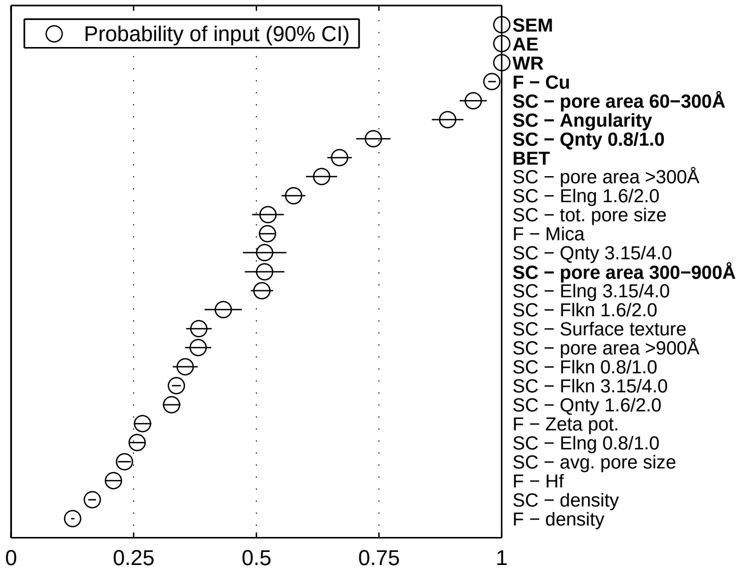

Figure 4.8: Concrete quality estimation example, predicting the air-% with MLP: The marginal posterior probabilities of inputs with a Beta-bin(27,5,10) prior on the number of inputs. The nine most probable inputs in the GP case are in boldface (see Figure 4.3).

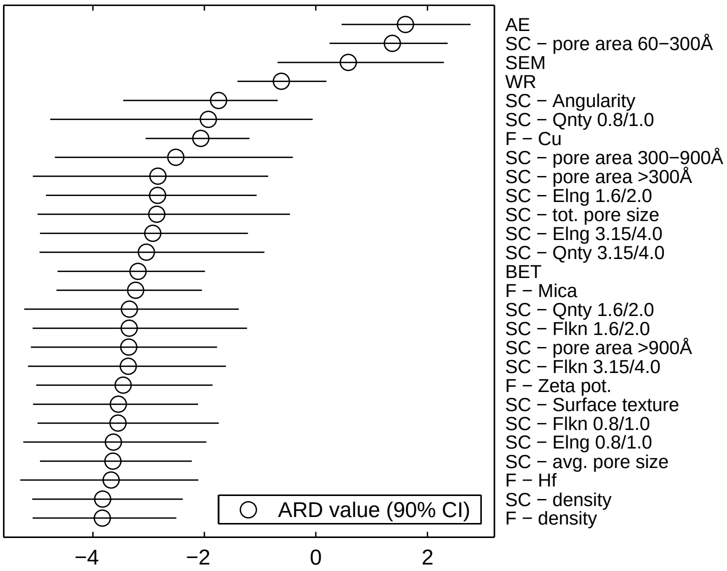

Figure 4.9: Concrete quality estimation example, predicting the air-% with MLP: The ARD values of the inputs of the full model. Compare to Figure 4.8.

between more probable inputs and less probable inputs. The nine most probable inputs are same as in the GP case (compare to Figure 4.3), except that "SC - pore area >300Å" has replaced very similar input "SC - pore area >300-900Å". Figure 4.9 shows the ARD values of the inputs for the full model. The order of the inputs based on the ARD values is clearly different from the order of the inputs based on marginal posterior probabilities (compare to Figure 4.8). Figure 4.10 shows the posterior probabilities of the ten most probable input combinations with a Beta-bin(27,5,10) prior on the number of inputs. There is more variation in the input combinations than in the case of GP and no model is significantly more probable than the others (compare to Figure 4.5).

Although different inputs would be selected in the case of MLP from the case of GP, the predictive performance (measured with the cross-validation predictive likelihood) was similar for both model types. Figure 4.11 shows the expected utilities computed by using the cross-validation predictive densities for the full model and the models having the k ($k = 5, \ldots, 15$) most probable inputs. The cross-validation predictive likelihoods are similar for the models having at least about eight most probable inputs and similar to the cross-validation predictive likelihoods of the GP models (Figure 4.6).

To illustrate the variation depending on the target variable, we present next the results for the GP model predicting the bleeding (exactly the same model and

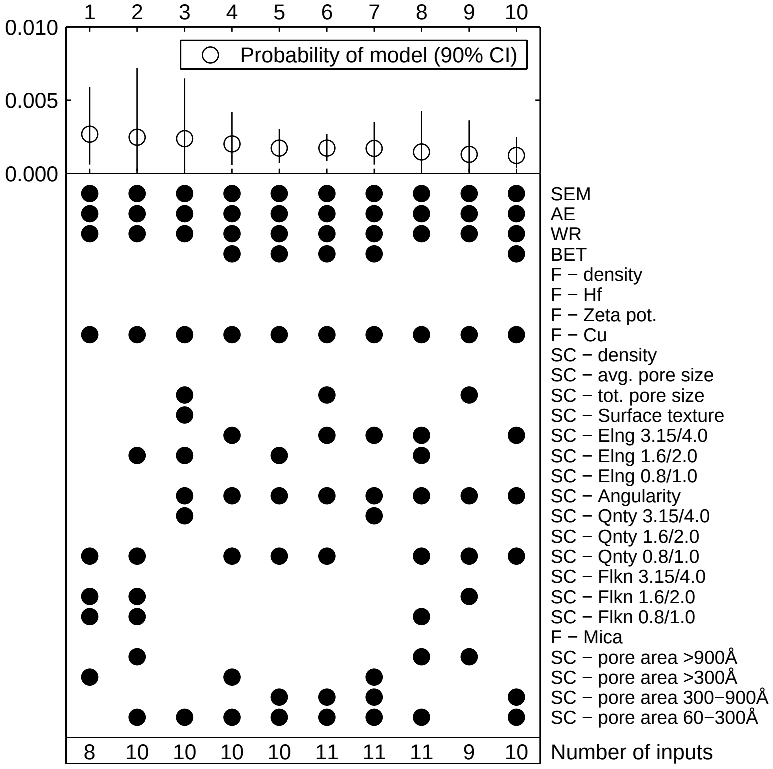

Figure 4.10: Concrete quality estimation example, predicting the air-% with MLP: The probabilities of the ten most probable models with a Beta-bin(27,5,10) prior on the number of inputs. The top part shows the probabilities of the models, the middle part shows which inputs are in the model, and the bottom part shows the number of inputs in the model. Compare to the results for the GP in Figure 4.5.

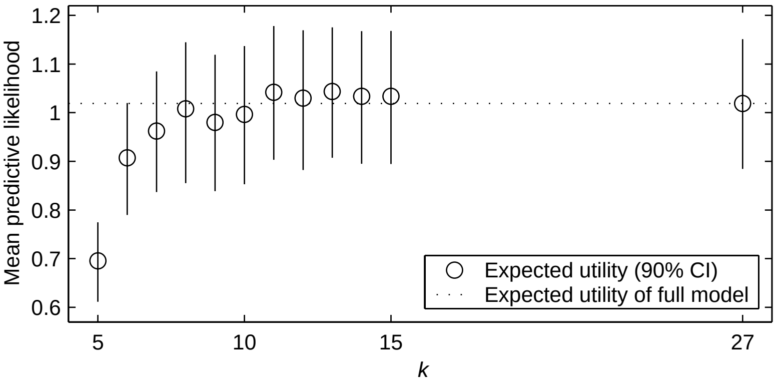

Figure 4.11: Concrete quality estimation example, predicting the air-% with MLP: The expected utilities of the models having the *k* most probable inputs (see Figure 4.8). Compare to results for the GP in Figure 4.6.

prior specification was used as in the GP model predicting the air-%). Figure 4.12 shows the marginal posterior probabilities of the number of inputs and Figure 4.13 shows the posterior probabilities of the ten most probable models. About one half of the inputs have higher posterior probability than the mean prior probability (1/3). The probability mass has been spread to many inputs and many similar models because of many correlating inputs. It is less clear than in the case of air-%, which are the most probable inputs and input combinations. However, as the most probable models had indistinguishable expected utilities, there was no danger of selecting a bad model. Note how the input *"SC-Qnty 0.8/1.0"*, which is included in the most probable model, has lower marginal probability than the five other inputs not in that model. This is not peculiar as the five particular inputs correlate strongly with the inputs in the most probable model.

In addition to using the cross-validation predictive likelihoods for model selection, we also computed the expected 90%-quantiles of absolute errors. These were used to confirm that there was no practical difference in prediction accuracy between the few most probable models. Naturally, it was also very important to report to the concrete expert the goodness of the models using easily understandable terms.

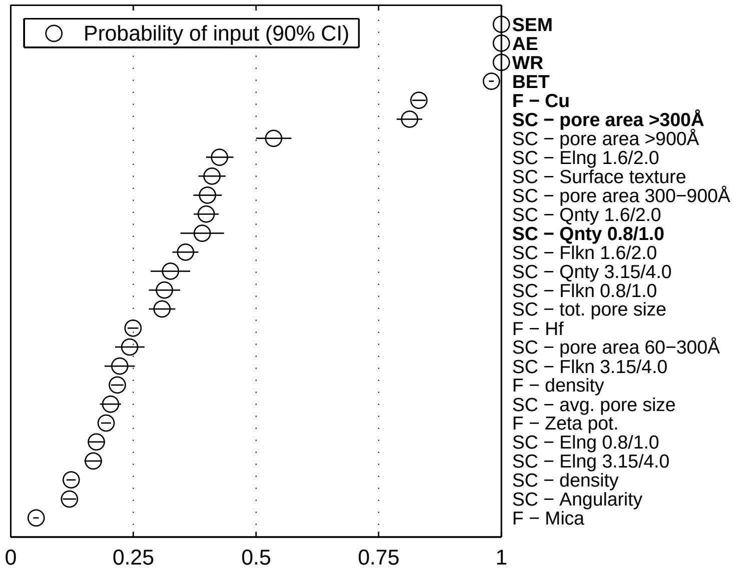

Figure 4.12: Concrete quality estimation example, predicting the bleeding with GP: The marginal posterior probabilities of inputs with a Beta-bin(27,5,10) prior on the number of inputs. The inputs in the most probable model are in boldface (see Figure 4.13).

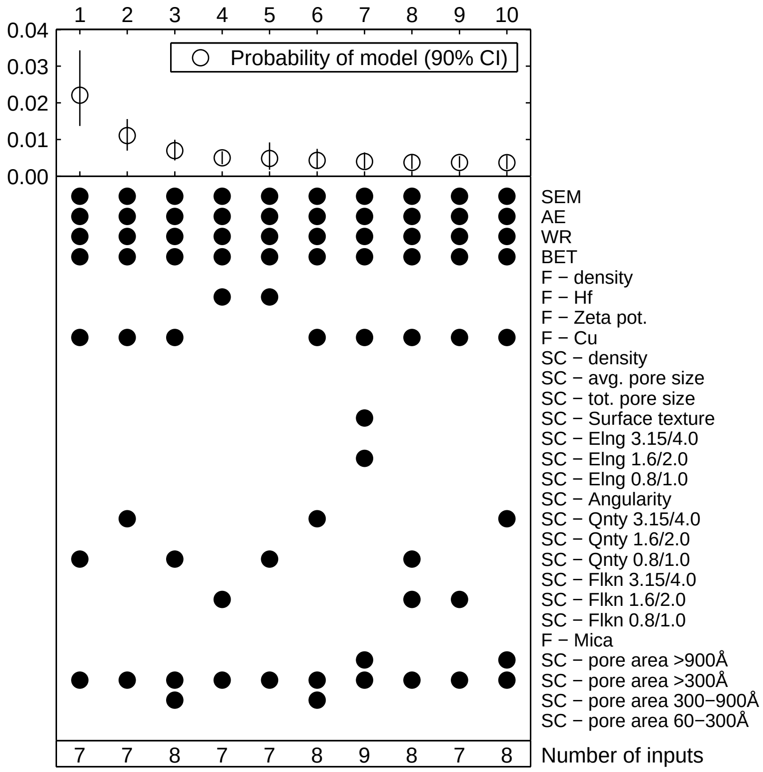

Figure 4.13: Concrete quality estimation example, predicting the bleeding with GP: Probabilities of the ten most probable models with a Beta-bin(27,5,10) prior on the number of inputs.

4.3.2 Case II: Forest scene classification

In this section, we illustrate that in more complex problems it may be necessary to aid input selection by using the marginal probabilities of the inputs.

The case problem is the classification of forest scenes with MLP (described in section 2.3.2). The primary goal was to check if these features contained enough information to produce reasonable classification results (see section 2.3.2) and the secondary goal was to reduce the computational burden by reducing the number of features used for the classification.

We used a 20-hidden-unit MLP with the logistic likelihood model. The between-model jump acceptance rates were about 2% and 0.4% with uniform and Beta priors, respectively. To increase the effective jump acceptance, between-model jumps were made three and nine times more probable, respectively, than in-model jumps.

From about $2 \cdot 10^{25}$ possible input combinations, the 4000 saved states visited about 3700 and 2500 different input combinations with uniform and Beta priors, respectively. None of the ten independent chains visited any input combination visited by the other chains. Consequently, it was impossible to make good estimates of the probabilities of the input combinations. Instead of trying to obtain an enormous amount of samples, it was possible to choose potentially useful input combinations by using the marginal posterior probabilities of inputs.

Figure 4.14 shows the posterior probabilities of the number of inputs with equal prior probability for all models. Due to the implicit Binomial prior on the number of inputs (see discussion in section 4.2.1), the probability mass is concentrated between 30 to 54 inputs. Figure 4.15 shows the posterior probabilities of the number of inputs with a Beta-bin(84,5,15) prior on the number of inputs favoring smaller models. The RJMCMC did not generate samples from models having fewer than 24 inputs (compare to the cross-validation predictive results in Figure 4.17), but this may have been caused by poor between-model convergence when the number of inputs was less than 30. The poor between-model convergence was identified by convergence diagnostics, and it seemed very unlikely that better results could have been obtained in reasonable time.

As the results with a uniform prior on the models had reasonable convergence, it was possible to estimate the relative importance of the inputs using the marginal posterior probabilities of the inputs from that run (Figure 4.16). Figure 4.17 shows the comparison of the expected utilities of the models having the *k* most probable inputs (*k* between 10 and 40). Reasonable results were achieved also with models having fewer inputs than the smallest model in the RJMCMC. Based on classification error results, just 12 (1/7th of the inputs in the full model) inputs would be sufficient in the planned application. Note that the difference in the performance between the 12 input model and full model is statistically but not practically significant.

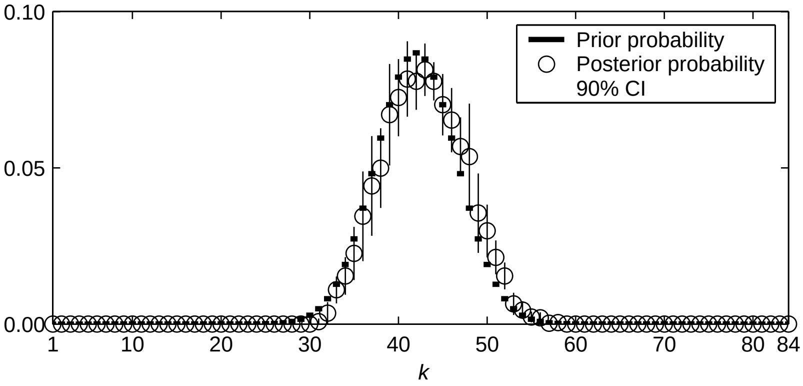

Figure 4.14: Forest scene classification example: The posterior probabilities of the number of inputs with a uniform prior on the models. The posterior probabilities are similar to prior probabilities and the probability mass is concentrated between 30 and 54 inputs.

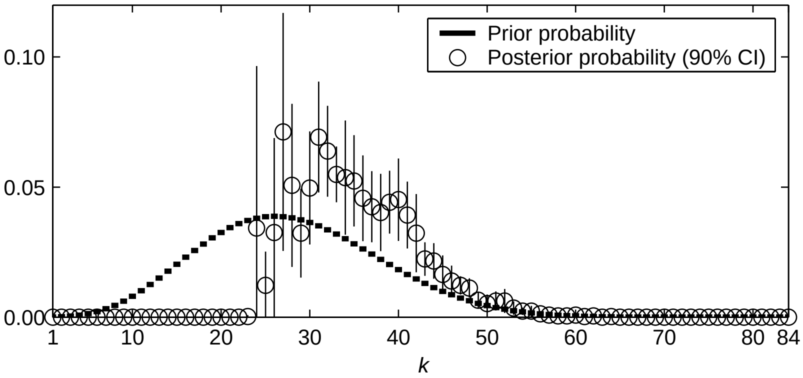

Figure 4.15: Forest scene classification example: The posterior probabilities of the number of inputs with a Beta-Bin(84,5,15) prior on the number of inputs. The poor between-model convergence can also be noticed from the large uncertainties in the probability estimates seen in this figure (compare to Figure 4.14).

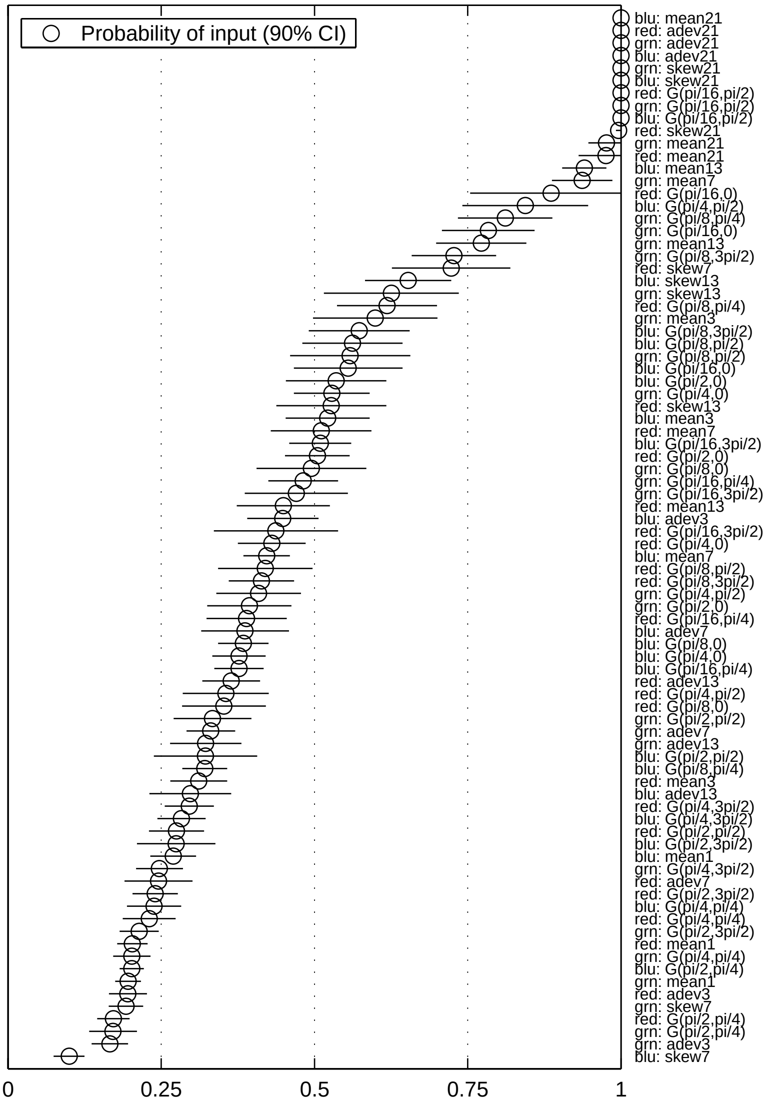

Figure 4.16: Forest scene classification example: The marginal posterior probabilities of the inputs with a uniform prior on models. These probabilities can be used to estimate the relevance of the inputs.

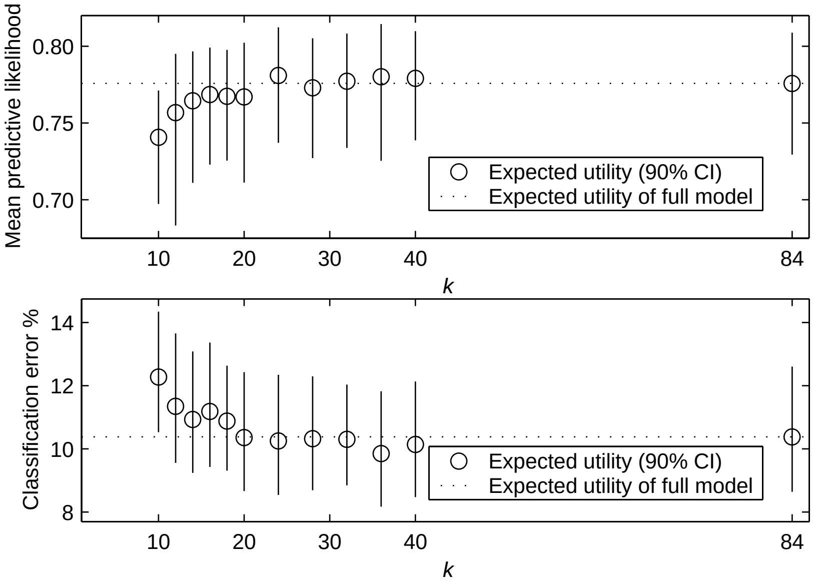

Figure 4.17: Forest scene classification example: The expected utilities of the models having the *k* most probable inputs (see Figure 4.16). The top plot shows the mean predictive likelihood and the bottom plot shows the classification error. The performance starts decreasing when there are fewer than about 20 inputs (compare to the RJMCMC results in Figure 4.15).

Chapter 5

Conclusions

The important advantage of the Bayesian approach is the possibility to handle the situation where some of the prior knowledge is lacking or vague, so that one is not forced to guess values for attributes that are unknown. For example, we do not need to guess in advance the number of degrees of freedom in the models, the distribution of model residuals, or the degree of complexity (nonlinearity) of the model with respect to each input variable. However, the results of any data analysis depend on the assumptions and approximations made – thus the Bayesian approach does not automatically give better results than any other approach. Even though the Bayesian models do not need validation data to set the model complexity, the validation of the final model is essential, which is the case also with any other modeling approach.

We presented how to compute the distribution of the expected utility, which can be used to describe, in terms of the application field, the goodness of the predictive ability of a Bayesian model and the uncertainty in that estimate. The IS-LOO-CV predictive densities are a quick way to estimate the expected utilities and the approach is also useful in some cases with flexible nonlinear models such as MLP and GP. If diagnostics hint that importance weights are not good, we can instead use the k-fold-CV predictive densities with the bias correction. Using the k-fold-CV takes k times more time, but it is more reliable. In addition, if data have certain dependencies, the k-fold-CV has to be used to get reasonable results. We proposed a quick and generic approach based on the Bayesian bootstrap for obtaining samples from the distributions of the expected utilities. With the proposed method, it is also easy to compute the probability that one model has better expected utility than another one. We discussed the assumptions and restrictions in the approach and relations to approaches for comparison of methods, other predictive densities, Bayes factors, information criteria, and the effective number of parameters. We also demonstrated how the cross-validation approach can be used to estimate the effective number of parameters. We illustrated the discussion and demonstrated the usefulness of the approach using one toy problem and two real world problems.

It might be useful in the future to study the importance link functions to improve the importance weights in IS-LOO-CV (section 3.2.1) and the coupling of the Markov chains to reduce the computation time in k-fold-CV (section 3.2.3). In addition, it would be useful to further investigate the relation to the DIC and the effective number of parameters (section 3.3.4). Specifically, it would be useful to study in which cases the DIC approach can be used to obtain reasonable estimates of the expected utilities and how to diagnose potential problems.

In the case of input variable selection, our goal was to select a smaller set of input variables in order to make the model more explainable and to reduce the cost of making measurements and the cost of computation. In addition, if the assumptions of the model and prior do not match well the properties of the data, reducing the number of input variables may improve the performance of the model. We proposed to use a variable dimension MCMC method to find out potentially useful input combinations and to do the final model choice and assessment using the expected utilities (with any desired utility) computed by using the cross-validation predictive densities. We discussed briefly the RJMCMC method, which is one of the simplest and fastest variable dimension MCMC methods.s The approach is based on the fact that the posterior probabilities of the model, given by the RJMCMC, are proportional to the product of the prior probabilities of the models and the prior predictive likelihoods of the models, which can be used to estimate the lower limit of the expected cross-validation predictive likelihood. We discussed different ways of including information about prior probabilities on the number of input variables. Additionally, in the case of very many inputs, we proposed that instead of using the probabilities of input combinations, the marginal probabilities of inputs can be used to select potentially useful models. We illustrated the discussion and demonstrated the usefulness of the approach using two real world problems.

Mixing speed of RJMCMC depends on proposal distributions. Use of conditional priors as proposal distributions worked reasonable well. Although we could not improve mixing with advanced approaches, further study might produce better alternatives. Furthermore, it might be useful to study such hierarchical priors for inputs that would allow posterior analysis of hierarchical relations of the input variables.

To summarize the results of this thesis, expected utilities are a useful way to assess and compare Bayesian models and good estimates of the expected utilities can be computed using cross-validation methods. Main contributions of the work are the theoretical and methodological advances, which provide solid framework to assess the performance of the models in the terms of the application specialist, while taking properly into account the uncertainties in the process.

References

Aitkin, M. (1991). Posterior Bayes factors (with discussion). *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(1):111–142.

Akaike, H. (1969). Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics*, 21:243–247.

Akaike, H. (1970). Statistical predictor identification. *Annals of the Institute of Statistical Mathematics*, 22:203–217.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In Petrov, B. N. and Csaki, F., editors, *Second International Symposium on Information Theory*, pp. 267–281, Budapest. Academiai Kiado. Reprinted in Kotz, S. and Johnson, N. L., editors, (1992). *Breakthroughs in Statistics Volume I: Foundations and Basic Theory*, pp. 610–624. Springer-Verlag.

Akaike, H. (1981). Likelihood of a model and information criteria. *Journal of Econometrics*, 16:3–14.

Andrieu, C., de Freitas, J. F. G., and Doucet, A. (2000). Robust full Bayesian methods for neural networks. In Solla, S. A., Leen, T. K., and Müller, K.-R., editors, *Advances in Neural Information Processing Systems 12*, pp. 379–385. MIT Press.

Barber, D. and Bishop, C. M. (1998). Ensamble learning in Bayesian neural networks. In Bishop, C. M., editor, *Neural Networks and Machine Learning*, pp. 215–237. Springer-Verlag.

Barber, D. and Williams, C. K. I. (1997). Gaussian process for Bayesian classification via hybrid Monte Carlo. In Mozer, M. C., Jordan, M. I., and Petsche, T., editors, *Advances in Neural Information Processing Systems 9*. MIT Press.

Barnard, J., McCulloch, R., and Meng, X.-L. (2000). Modelling covariance matrices in terms of standard deviations and correlations with application to shrinkage. *Statistica Sinica*, 10(4):1281–1312.

Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances. *The Philosophical Transactions*, 53:370–418. Reprinted 1958 in *Biometrika*, 45(3/4):296–315.

Berger, J. and Pericchi, L. (1996). The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91(433):109–122.

Berger, J. O. (1985). *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, 2nd edition.

Berger, J. O. and Bernardo, J. M. (1992). On the development of reference priors. In Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics 4*, pp. 35–60. Oxford University Press.

Bernardo, J. M. (1979). Expected information as expected utility. *Annals of Statistics*, 7(3):686–690.

Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory*. John Wiley & Sons.

Bishop, C. M. (1993). Curvature-driven smoothing: A learning algorithm for feed-forward networks. *IEEE Transactions on Neural Networks*, 4(5):882–884.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

Bishop, C. M. and Qazaz, C. S. (1997). Regression with input-dependent noise: A Bayesian treatment. In Mozer, M. C., Jordan, M. I., and Petsche, T., editors, *Advances in Neural Information Processing Systems 9*, pp. 347–353. MIT Press.

Box, G. E. P. (1980). Sampling and Bayes' inference in scientific modelling and robustness. *Journal of the Royal Statistical Society. Series A (General)*, 143(4):383–430.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Chapman and Hall.

Brooks, S. P. and Giudici, P. (1999). Convergence assessment for reversible jump MCMC simulations. In Bernardo, J. M., Berger, J. O., and Dawid, A. P., editors, *Bayesian Statistics 6*, pp. 733–742. Oxford University Press.

Brooks, S. P. and Giudici, P. (2000). Markov chain Monte Carlo convergence convergence assessment via two-way analysis of variance. *Journal of Computational and Graphical Statistics*, 9(2):266–285.

Brooks, S. P., Giudici, P., and Philippe, A. (2001). Nonparametric convergence assessment for MCMC model selection [online]. Updated 2001-03-06. Available at http://www.statslab.cam.ac.uk/~steve/mypapers/brogp01.ps.

Brooks, S. P., Giudici, P., and Roberts, G. O. (2000). Efficient construction of reversible jump MCMC proposal distributions [online]. Updated 2000-12-06. Available at http://www.statslab.cam.ac.uk/~steve/mypapers/brogr00.ps.

Brooks, S. P. and Roberts, G. O. (1999). Assessing convergence of Markov chain Monte Carlo algorithms. *Statistics and Computing*, 8(4):319–335.

Buntine, W. L. and Weigend, A. S. (1991). Bayesian back-propagation. *Complex systems*, 5(6):603–643.

Burman, P. (1989). A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, 76(3):503–514.

Burman, P., Chow, E., and Nolan, D. (1994). A cross-validatory method for dependent data. *Biometrika*, 81(2):351–358.

Burman, P. and Nolan, D. (1992). Data dependent estimation of prediction functions. *Journal of Time Series Analysis*, 13(3):189–207.

Burnham, K. P. and Anderson, D. R. (1998). *Model selection and inference*. Springer.

Carlin, B. P. and Chib, S. (1995). Bayesian model choice via Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(3):473–484.

Chen, M.-H., Shao, Q.-M., and Ibrahim, J. Q. (2000). *Monte Carlo Methods in Bayesian Computation*. Springer-Verlag.

Chipman, H. (1996). Bayesian variable selection with related predictors. *Canadian Journal of Statistics*, 24(1):17–36.

Chipman, H., George, E. I., and McCulloch, R. E. (2001). Practical implementation of Bayesian model selection [online]. Updated July 4th 2001. Available at http://bevo2.bus.utexas.edu/georgee/Research\%20papers/Practical.pdf.

Chow, G. C. (1981). A comparison of the information and posterior probability criteria for model selection. *Journal of Econometrics*, 16:21–33.

Cox, R. T. (1946). Probability, frequency and reasonable expectation. *American Journal of Physics*, 14(1):1–13.

Csató, L., Fokoué, E., Opper, M., Schottky, B., and Winther, O. (2000). Efficient approaches to Gaussian process classification. In Solla, S. A., Leen, T. K., and Müller, K.-R., editors, *Advances in Neural Information Processing Systems 12*, pp. 251–257. MIT Press.

de Freitas, J. F. G., Niranjan, M., Gee, A. H., and Doucet, A. (2000). Sequential Monte Carlo methods to train neural network models. *Neural Computation*, 12(4):955–993.

Dellaportas, P. and Forster, J. J. (1999). Markov chain Monte Carlo model determination for hierarchical and graphical log-linear models. *Biometrika*, 86(3):615–633.

Denison, D. G. T., Mallick, B. K., and Smith, A. F. M. (1998). Automatic Bayesian curve fitting. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 60(2):333–350.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1924.

Draper, D. (1995a). Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):45–97.

Draper, D. (1995b). Model uncertainty, data mining and statistical inference: Discussion. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 158(3):450–451.

Draper, D. (1996). Posterior predictive assessment of model fitness via realized discrepancies: Discussion. *Statistica Sinica*, 6(4):760–767.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222.

Findley, D. F. and Parzen, E. (1995). A conversation with Hirotugu Akaike. *Statistical Science*, 10(1):104–117.

Gamerman, D. (1997). *Markov Chain Monte Carlo: Stochastic simulation for Bayesian inference*. Chapman & Hall.

Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70(350):320–328.

Geisser, S. and Eddy, W. F. (1979). A predictive approach to model selection. *Journal of the American Statistical Association*, 74(365):153–160.

Gelfand, A. E. (1996). Model determination using sampling-based methods. In Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., editors, *Markov Chain Monte Carlo in Practice*, pp. 145–162. Chapman & Hall.

Gelfand, A. E. and Dey, D. K. (1994). Bayesian model choice: Asymptotics and exact calculations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(3):501–514.

Gelfand, A. E., Dey, D. K., and Chang, H. (1992). Model determination using predictive distributions with implementation via sampling-based methods (with discussion). In Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics 4*, pp. 147–167. Oxford University Press.

Gelman, A. (1996). Inference and monitoring convergence. In Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., editors, *Markov Chain Monte Carlo in Practice*, pp. 131–144. Chapman & Hall.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. R. (1995). *Bayesian Data Analysis*. Chapman & Hall.

Gelman, A., Goegebeur, Y., Tuerlinckx, F., and Van Mechelen, I. (2000). Diagnostic checks for discrete data regression models using posterior predictive simulations. *Journal of the Royal Statistical Society. Series C (Applied statistics)*, 49(3):247–268.

Gelman, A. and Meng, X.-L. (1996). Model checking and model improvement. In Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., editors, *Markov Chain Monte Carlo in Practice*, pp. 189–202. Chapman & Hall.

Gelman, A., Meng, X.-L., and Stern, H. (1996). Posterior predictive assessment of model fitness via realized discrepancies (with discussion). *Statistica Sinica*, 6(4):733–807.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):721–741.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57(6):1317–1339.

Geweke, J. (1993). Bayesian treatment of the independent Student-*t* linear model. *Journal of Applied Econometrics*, 8(Supplement):S19–S40.

Gibbs, M. N. (1997). *Bayesian Gaussian Processes for Regression and Classification*. PhD thesis, University of Cambridge.

Gilks, W. R. (1995). Fractional Bayes factors for model comparison: Discussion. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):119.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., editors (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall.

Gilks, W. R., Thomas, A., and Spiegelhalter, D. J. (1992). A language and program for complex Bayesian modelling. *Statistician*, 43(1):169–177.

Goel, P. K. and Degroot, M. H. (1981). Information about hyperparameters in hierarchical models. *Journal of the American Statistical Association*, 76(373):140–147.

Goldberg, P. W., Williams, C. K. I., and Bishop, C. M. (1998). Regression with input-dependent noise: A Gaussian process treatment. In Jordan, M. I., Kearns, M. J., and Solla, S. A., editors, *Advances in Neural Information Processing Systems 10*. MIT Press.

Good, I. J. (1952). Rational decisions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 14(1):107–114.

Green, P. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732.

Han, C. and Carlin, B. P. (2001). Markov chain Monte Carlo methods for computing Bayes factors: A comparative review. *Journal of the American Statistical Association*, 96(455):1122–1132.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Hill, B. M. (1982). Lindley's paradox: Comment. *Journal of the American Statistical Association*, 77(378):344–347.

Hurvich, C. M. and Tsai, C.-L. (1989). Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307.

Hurvich, C. M. and Tsai, C.-L. (1991). Bias of the corrected aic criterion for underfitted regression and time series models. *Biometrika*, 78(3):499–509.

Jaynes, E. T. (1996). *Probability Theory: The Logic of Science [online]*. Fragmentary edition of March 1996. Available at http://bayes.wustl.edu/etj/prob.html.

Jeffreys, H. (1961). *Theory of Probability*. Oxford University Press, 3rd edition. (1st edition 1939).

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1998). An introduction to variational methods for graphical models. In Jordan, M. I., editor, *Learning in Graphical Models*. Kluwer Academic Publishers.

Järvenpää, H. (2001). *Quality characteristics of fine aggregates and controlling their effects on concrete*. Acta Polytechnica Scandinavica, Civil Engineering and Building Construction Series No. 122. The Finnish Academy of Technology. Dissertation, Helsinki University of Technology, Finland.

Kaipio, J. P., Kolehmainen, V., Somersalo, E., and Vauhkonen, M. (2000). Statistical inversion and Monte Carlo sampling methods in electrical impedance tomography. *Inverse Problems*, 16(5):1487–1522.

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795.

Kass, R. E. and Wasserman, L. (1996). The selection of prior distributions by formal rules. *Journal of the American Statistical Association*, 91(435):1343–1370.

Kitagawa, G. (1987). Non-gaussian state-space modeling of nonstationary time series: Rejoinder. *Journal of the American Statistical Association*, 82(400):1060–1063.

Künsch, H. R. (1989). The jackknife and the bootstrap for general stationary observations. *Annals of Statistics*, 17(3):1217–1241.

Künsch, H. R. (1994). Approximate Bayesian inference with the weighted likelihood bootstrap: Discussion. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(1):39.

Kohn, R., Smith, M., and Chan, D. (2001). Nonparametric regression using linear combination of basis functions. *Statistics and Computing*, 11(4):313–322.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288.

Konishi, S. and Kitagawa, G. (1996). Generalized information criteria in model selection. *Biometrika*, 83(4):875–890.

Krogh, A. and Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. In Tesauro, G., Touretzky, D. S., and Leen, T. K., editors, *Advances in Neural Information Processing Systems 7*, pp. 231–238. MIT Press.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86.

Lampinen, J. and Selonen, A. (1997). Using background knowledge in multilayer perceptron learning. In Frydrych, M., Parkkinen, J., and Visa, A., editors, *SCIA'97: Proceedings of the 10th Scandinavian Conference on Image Analysis*, volume 2, pp. 545–549. Pattern Recognition Society of Finland.

Lampinen, J. and Vehtari, A. (2001). Bayesian approach for neural networks – review and case studies. *Neural Networks*, 14(3):7–24.

Lampinen, J., Vehtari, A., and Leinonen, K. (1999). Using Bayesian neural network to solve the inverse problem in electrical impedance tomography. In Ersboll, B. K. and Johansen, P., editors, *SCIA'99: Proceedings of the 11th Scandinavian Conference on Image Analysis*, volume 1, pp. 87–93. The Pattern Recognition Society of Denmark.

Laplace, P. S. (1774). Mémoire sur la probabilité des causes par les évènemens. *Mémoires de Mathématique et de Physique, Presentés à l'Académie Royale des Sciences, par divers Savans & lûs dans ses Assemblées, Tome Sixième*, pp. 621–656. English translation by S. M. Stigler: Laplace, P. S. (1986). Memoir on the probability of the causes of events. *Statistical Science*, 1(3):364–478.

Laplace, P. S. (1825). *Essai philosophique sur les probabilités*. Bachelier, Paris, 5th edition. English translation with notes by A. I. Dale: Laplace, P. S. (1995). *Philosophical Essay on Probabilities*. Springer-Verlag.

Lemm, J. C. (1996). Prior information and generalized questions. Technical Report AIM 1598, CBCLP 141, Massachusetts Institute of Technology, Artificial Intelligence Laboratory and Center for Biological and Computational Learning, Department of Brain and Cognitive Sciences.

Lemm, J. C. (1999). Bayesian field theory. Technical Report MS-TP1-99-1, Universität Münster, Institut für Theoretische Physik.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag.

Liu, J. S. and Chen, R. (1995). Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90(430):567–576.

Lo, A. Y. (1987). A large sample study of the Bayesian bootstrap. *Annals of Statistics*, 15(1):360–375.

MacEachern, S. N. and Peruggia, M. (2000). Importance link function estimation for Markov chain Monte Carlo methods. *Journal of Computational and Graphical Statistics*, 9(1):99–121.

MacKay, D. J. C. (1992). A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472.

MacKay, D. J. C. (1994). Bayesian non-linear modelling for the prediction competition. In *ASHRAE Transactions, V.100, Pt.2*, pp. 1053–1062. ASHRAE.

MacKay, D. J. C. (1995). Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6(3):469–505.

MacKay, D. J. C. (1998a). Introduction to Gaussian processes. In Bishop, C. M., editor, *Neural Networks and Machine Learning*, pp. 133–165. Springer-Verlag.

MacKay, D. J. C. (1998b). Introduction to Monte Carlo methods. In Jordan, M. I., editor, *Learning in Graphical Models*. Kluwer Academic Publishers.

Mason, D. M. and Newton, M. A. (1992). A rank statistics approach to the consistency of a general bootstrap. *Annals of Statistics*, 20(3):1611–1624.

Metropolis, N., Rosenbluth, A., Rosenbluth, R., Teller, A., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092.

Müller, P. and Rios Insua, D. (1998). Issues in Bayesian analysis of neural network models. *Neural Computation*, 10(3):571–592.

Moody, J. E. (1992). The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In Moody, J. E., Hanson, S. J., and Lippmann, R. P., editors, *Advances in Neural Information Processing Systems 4*, pp. 847–854. Morgan Kaufmann Publishers.

Morgan, N. and Bourland, H. (1990). Generalization and parameter estimation in feedforward nets: Some experiments. In Touretzky, D., editor, *Advances in Neural Information Processing Systems 2*, pp. 630–637. Morgan Kaufman.

Murata, N., Yoshizawa, S., and Amari, S.-I. (1994). Network information criterion—determining the number of hidden units for an artificial neural network model. *IEEE Transactions on Neural Networks*, 5(6):865–872.

Nadeau, C. and Bengio, S. (2000). Inference for the generalization error. In Solla, S. A., Leen, T. K., and Müller, K.-R., editors, *Advances in Neural Information Processing Systems 12*, pp. 307–313. MIT Press.

Neal, R. M. (1992). Bayesian training of backpropagation networks by the hybrid Monte Carlo method. Technical Report CRG-TR-92-1, Dept. of Computer Science, University of Toronto.

Neal, R. M. (1993). Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto.

Neal, R. M. (1994). An improved acceptance procedure for the Hybrid Monte Carlo algorithm. *Journal of Computational Physics*, 111(1):194–203.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Springer-Verlag.

Neal, R. M. (1997). Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical Report 9702, Dept. of Statistics, University of Toronto.

Neal, R. M. (1998). Assessing relevance determination methods using DELVE. In Bishop, C. M., editor, *Neural Networks and Machine Learning*, pp. 97–129. Springer-Verlag.

Neal, R. M. (1999). Regression and classification using Gaussian process priors (with discussion). In Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics 6*, pp. 475–501. Oxford University Press.

Newton, M. A. and Raftery, A. E. (1994). Approximate Bayesian inference with the weighted likelihood bootstrap (with discussion). *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(1):3–48.

Ntzoufras, I. (1999). *Aspects of Bayesian model and variable selection using MCMC*. PhD thesis, Department of Statistics, Athens University of Economics and Business.

O'Hagan, A. (1995). Fractional Bayes factors for model comparison (with discussion). *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):99–138.

Orr, M. J. L. (1996). Introduction to radial basis function networks [online]. Technical report, Centre for Cognitive Science, University of Edinburgh. April 1996. Available at http://www.anc.ed.ac.uk/~mjo/papers/intro.ps.gz.

Peruggia, M. (1997). On the variability of case-deletion importance sampling weights in the Bayesian linear model. *Journal of the American Statistical Association*, 92(437):199–207.

Phillips, D. B. and Smith, A. F. M. (1996). Bayesian model comparison via jump diffusions. In Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., editors, *Markov Chain Monte Carlo in Practice*, pp. 215–239. Chapman & Hall.

Pinto, R. L. and Neal, R. M. (2001). Improving Markov chain Monte Carlo estimators by coupling to an approximating chain. Technical Report 0101, Dept. of Statistics, University of Toronto.

Poncet, A. (1996). Asymptotic probability density of the generalization error. In *Proceedings of International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing*, pp. 66–74. IEEE.

Prechelt, L. (1998). Early stopping – but when? In Orr, G. B. and Müller, K.-R., editors, *Neural Networks: Tricks of the Trade*, volume 1524 of *Lecture Notes in Computer Science*, pp. 55–69. Springer-Verlag.

Rasmussen, C. E. (1996). *Evaluation of Gaussian Processes and other Methods for Non-Linear Regression*. PhD thesis, Department of Computer Science, University of Toronto.

Rasmussen, C. E., Neal, R. M., Hinton, G. E., van Camp, D., Revow, M., Ghahramani, Z., Kustra, R., and Tibshirani, R. (1996). The DELVE manual [online]. Version 1.1. Available at ftp://ftp.cs.utoronto.ca/pub/neuron/delve/doc/manual.ps.gz.

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society. Series B (Methodological)*, 59(4):731–792.

Rios Insua, D. and Müller, P. (1998). Feedforward neural networks for nonparametric regression. In Dey, D. K., Müller, P., and Sinha, D., editors, *Practical Nonparametric and Semiparametric Bayesian Statistics*, pp. 181–194. Springer-Verlag.

Robert, C. P. and Casella, G. (1999). *Monte Carlo Statistical Methods*. Springer-Verlag.

Rubin, D. B. (1981). The Bayesian bootstrap. *Annals of Statistics*, 9(1):130–134.

Rubin, D. B. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *Annals of Statistics*, 12(4):1151–1172.

Sarle, W. S. (1997). How to measure importance of inputs? [online]. Technical report, SAS Institute Inc. Revised June 23, 2000. Available at ftp://ftp.sas.com/pub/neural/importance.html.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464.

Shao, J. (1993). Linear model selection by cross-validation. *Journal of the American Statistical Association*, 88(422):486–494.

Shibata, R. (1976). Selection of the order of an autoregressive model by Akaike's information criterion. *Biometrika*, 63(1):117–126.

Sivia, D. S. (1996). *Data Analysis: A Bayesian tutorial*. Oxford Science Publications.

Smola, A. J. and Bartlett, P. (2001). Sparse greedy Gaussian process regression. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13*, pp. 619–625. MIT Press.

Spiegelhalter, D., Thomas, A., Best, N., and Gilks, W. (1996). *BUGS 0.5 * Examples Volume 1 (version i)*. MRC Biostatistics Unit, Institute of Public Health.

Spiegelhalter, D. J. (1995). Assessment and propagation of model uncertainty: Discussion. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):71–73.

Spiegelhalter, D. J., Best, N. G., and Carlin, B. P. (1998). Bayesian deviance, the effective number of parameters, and the comparison of arbitrarily complex models. Research report 98-009, Division of Biostatistics, University of Minnesota.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and van der Linde, A. (2001). Bayesian measures of model complexity and fit. Research report 2001-013, Division of Biostatistics, University of Minnesota.

Spiegelhalter, D. J., Myles, J. P., Jones, D. R., and Abrams, K. R. (2000). Bayesian methods in health technology assessment: A review. *Health Technology Assessment 2000*, 4(38).

Stephens, M. (2000). Bayesian analysis of mixtures with an unknown number of components — an alternative to reversible jump methods. *Annals of Statistics*, 28(1):40–74.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions (with discussion). *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147.

Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):44–47.

Sugiyama, M. and Ogawa, H. (2000). Another look at C_p and network information criterion as approximation of subspace information criterion. Technical Report TR00-0012, Department of Computer Science, Tokyo Institute of Technology.

Sugiyama, M. and Ogawa, H. (2001). Subspace information criterion for model selection. *Neural Computation*, 13(8):1863–1889.

Sundararajan, S. and Keerthi, S. S. (2001). Predictive approaches for choosing hyperparameters in Gaussian processes. *Neural Computation*, 13(5):1103–1118.

Sykacek, P. (2000). On input selection with reversible jump Markov chain Monte Carlo sampling. In Solla, S. A., Leen, T. K., and Müller, K.-R., editors, *Advances in Neural Information Processing Systems 12*, pp. 638–644. MIT Press.

Trecate, G. F., Williams, C. K. I., and Opper, M. (1999). Finite-dimensional approximation of Gaussian processes. In Kearns, M. J., Solla, S. A., and Cohn, D. A., editors, *Advances in Neural Information Processing Systems 11*. MIT Press.

Vehtari, A., Heikkonen, J., Lampinen, J., and Juujärvi, J. (1998). Using Bayesian neural networks to classify forest scenes. In Casasent, D. P., editor, *Intelligent Robots and Computer Vision XVII: Algorithms, Techniques, and Active Vision*, pp. 66–73. SPIE.

Vehtari, A. and Lampinen, J. (1999a). Bayesian neural networks for industrial applications. In *SMCIA/99: Proceedings of the 1999 IEEE Midnight-Sun Workshop on Soft Computing Methods in Industrial Applications*, pp. 63–68. IEEE.

Vehtari, A. and Lampinen, J. (1999b). Bayesian neural networks with correlating residuals. In *IJCNN'99: Proceedings of the 1999 International Joint Conference on Neural Networks [CD-ROM]*, number 2061. IEEE.

Vehtari, A. and Lampinen, J. (2000). Bayesian MLP neural networks for image analysis. *Pattern Recognition Letters*, 21(13–14):1183–1191.

Vehtari, A. and Lampinen, J. (2001a). Bayesian input variable selection using cross-validation predictive densities and reversible jump MCMC. Technical Report B28, Helsinki University of Technology, Laboratory of Computational Engineering.

Vehtari, A. and Lampinen, J. (2001b). On Bayesian model assessment and choice using cross-validation predictive densities. Technical Report B23, Helsinki University of Technology, Laboratory of Computational Engineering.

Vehtari, A., Särkkä, S., and Lampinen, J. (2000). On MCMC sampling in Bayesian MLP neural networks. In Amari, S.-I., Giles, C. L., Gori, M., and Piuri, V., editors, *IJCNN'2000: Proceedings of the 2000 International Joint Conference on Neural Networks*, volume I, pp. 317–322. IEEE.

Vlachos, P. K. and Gelfand, A. E. (2000). On the calibration of Bayesian model choice criteria. Technical Report 705, Carnegie Mellon University, Department of Statistics.

Weng, C.-S. (1989). On a second-order asymptotic property of the Bayesian bootstrap mean. *Annals of Statistics*, 17(2):705–710.

Williams, C. K. I. and Rasmussen, C. E. (1996). Gaussian processes for regression. In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems 8*. MIT Press.

Williams, C. K. I. and Seeger, M. (2001). Using the Nyström method to speed up kernel machines. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13*, pp. 682–688. MIT Press.

Winther, O. (1998). *Bayesian Mean Field Algorithms for Neural Networks and Gaussian Processes*. PhD thesis, University of Copenhagen.

Wolpert, D. H. (1996a). The existence of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1391–1420.

Wolpert, D. H. (1996b). The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390.

Wolpert, D. W. and Macready, W. G. (1995). No free lunch theorems for search. Technical Report SFI-TR-95-02-010, The Santa Fe Institute.

Yang, R. and Berger, J. O. (1997). A catalog of noninformative priors. ISDS Discussion Paper 97-42, Institute of Statistics and Decision Sciences, Duke University.

Zhu, H., Williams, C. K. I., Rohwer, R., and Morciniec, M. (1998). Gaussian regression and optimal finite dimensional linear models. In Bishop, C. M., editor, *Neural Networks and Machine Learning*, pp. 167–184. Springer-Verlag.

Zhu, L. and Carlin, B. P. (2000). Comparing hierarchical models for spatio-temporally misaligned data using the deviance information criterion. *Statistics in Medicine*, 19(17–18):2265–2278.

ISBN 951-22-5764-5 (printed)
ISBN 951-22-5765-3 (PDF)
ISSN 1455-0474