The final version of this article will be published in
Neural Computation, Vol. 14, Issue 10, published by The MIT Press.

Bayesian Model Assessment and Comparison Using Cross-Validation Predictive Densities

Aki Vehtari and Jouko Lampinen
Laboratory of Computational Engineering
Helsinki University of Technology
P.O.Box 9400, FIN-02015, HUT, Finland
{Aki.Vehtari,Jouko.Lampinen}@hut.fi

March 7, 2002

Abstract

In this work, we discuss practical methods for the assessment, comparison, and selection of complex hierarchical Bayesian models. A natural way to assess the goodness of the model is to estimate its future predictive capability by estimating expected utilities. Instead of just making a point estimate, it is important to obtain the distribution of the expected utility estimate, as it describes the uncertainty in the estimate. The distributions of the expected utility estimates 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. We propose an approach using cross-validation predictive densities to obtain expected utility estimates and Bayesian bootstrap to obtain samples from their distributions. We also discuss the probabilistic assumptions made and properties of two practical cross-validation methods, importance sampling and k-fold cross-validation. As illustrative examples, we use MLP neural networks and Gaussian Processes (GP) with Markov chain Monte Carlo sampling in one toy problem and two challenging real-world problems.

Keywords: Bayesian methods; expected utility; cross-validation; model assessment; model selection; pseudo-Bayes factor; Monte Carlo; MLP neural networks; Gaussian processes

1 Introduction

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. 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, Dey, & Chang, 1992; Shao, 1993). The cross-validation predictive density dates at least to (Geisser & Eddy, 1979) and review of cross-validation and other predictive densities appears in (Gelfand & Dey, 1994; Gelfand, 1996). Bernardo and Smith (1994, chap. 6) also discuss briefly how cross-validation approximates the formal Bayes procedure of computing the expected utilities.

We synthesize and extend the previous work in many ways. We give a unified presentation from the Bayesian viewpoint emphasizing the assumptions made and propose practical methods to obtain the distributions of the expected utility estimates. We first review expected utilities (section 1.1) and cross-validation predictive densities (section 1.2) and discuss assumptions made on future data distribution (section 1.3). We discuss the properties of two practical methods, the importance sampling leave-one-out (section 2.1) and the k-fold cross-validation (section 2.2). We propose a quick and generic approach based on the Bayesian bootstrap for obtaining samples from the distributions of the expected utility estimates (section 2.3). If there is a collection of models under consideration, the distributions of the expected utility estimates can also be used for comparison (section 2.4). We also discuss the relation of the proposed method to prior predictive densities and Bayes factors (section 3.1), and posterior predictive densities (section 3.2).

As the estimation of the expected utilities requires a full model fitting (or k model fittings) for each model candidate, the 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 & Chib, 1995; Stephens, 2000) for model selection and still use the expected utilities for final model assessment. This approach is discussed in more detail by Vehtari (2001, chap. 4).

We have tried to follow the notation of Gelman, Carlin, Stern, and Rubin (1995) and we assume that the reader has basic knowledge of Bayesian methods (e.g., see a short introduction in (Lampinen & Vehtari, 2001)). To illustrate the discussion we use MLP networks and Gaussian processes (GP) with Markov chain Monte Carlo (MCMC) in one toy problem and two real world problems (section 4).

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 & 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

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

where θ denotes all the model parameters and hyperparameters of the prior structures and M is all the prior knowledge in the model specification (including all the 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 $p(y|x^{(n+1)}, \theta, D, M) = p(y|x^{(n+1)}, \theta, M)$.

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

$$u_h = u\left(y^{(n+h)}, x^{(n+h)}, D, M\right). \tag{2}$$

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

$$\bar{u} = E_h[u_h] \tag{3}$$

or an α-quantile

$$\bar{u}_\alpha = Q_{\alpha,h}[u_h]. \tag{4}$$

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

$$u_h = \left(E_y[y|x^{(n+h)}, D, M] - y^{(n+h)}\right)^2 \tag{5}$$

and the absolute error

$$u_h = \text{abs}\left(E_y[y|x^{(n+h)}, D, M] - y^{(n+h)}\right) \tag{6}$$

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

$$u_h = p(y^{(n+h)}|x^{(n+h)}, D, M) \tag{7}$$

and it is especially useful in model comparison (see section 2.4), and in non-prediction problems, in which the goal is just to get scientific insights in modeled phenomena. Maximization of the expected predictive likelihood corresponds to minimization of information-theoretic Kullback-Leibler (KL) divergence between the model and the unknown distribution of the data, and equivalently, it corresponds to maximization of the expected information gained (Bernardo, 1979).

An application specific utility may measure the expected benefit or cost. For simplicity we use term utility also for costs, although better word would be *risk*. Also, instead of negating cost, we represent the utilities in a form that is most appealing for the application expert. It should be obvious in each case whether a smaller or larger value for the utility is better.

1.2 Cross-validation predictive densities

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 $\{(x^{(i)}, y^{(i)}); i = 1, 2, \ldots, n\}$. 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

$$\{p(y|x^{(i)}, D^{(\setminus i)}, M); i = 1, 2, \ldots, n\}, \tag{8}$$

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

$$\bar{u}_{\mathrm{LOO}} = E_i[u(y^{(i)}, x^{(i)}, D^{(\setminus i)}, M)]. \tag{9}$$

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 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 2.2 and 4.3).

The LOO-CV-predictive densities are computed by the equation (compare to equation 1)

$$p(y|x^{(i)}, D^{(\setminus i)}, M) = \int p(y|x^{(i)}, \theta, D^{(\setminus i)}, M)\, p(\theta|D^{(\setminus i)}, M) d\theta. \qquad (10)$$

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 of the 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 2.1 or the k-fold-CV discussed in section 2.2 can be used to reduce the computational burden.

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 & Bengio, 2000, and references therein), where the goal is to compare (not assess) the performance of *methods (algorithms)* instead of the single *models* conditioned on the given training data.

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 1.1).

The method comparison approaches try to answer the question: "Given two methods A and B and training data D, which *method (algorithm)* 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 1)

$$p(y|x^{(n+h)}, D_h^*, M) = \int p(y|x^{(n+h)}, \theta, D_h^*, M)\, p(\theta|D_h^*, M) d\theta, \qquad (11)$$

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 theorem (Wolpert, 1996a,b). Even if the distributions Ω and Ω^+ have only few dimensions, it is very hard to quantify differences and estimate their effect on 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 absolute 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, Myles, Jones, & Abrams, 2000, pp. 18-19 and references therein).

2 Methods

In this section, we discuss the importance sampling leave-one-out (section 2.1) and k-fold cross-validation (section 2.2). We also propose an approach for obtaining samples from the distributions of the expected utility estimates (section 2.3) and discuss model comparison based on expected utilities (section 2.4).

2.1 Importance sampling leave-one-out cross-validation

In IS-LOO-CV, instead of sampling directly from $p(\theta|D^{(\setminus i)}, M)$, 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 $h(\theta)$ and we have samples $\dot{\theta}_j$ from distribution $g(\theta)$, we can write the expectation as

$$E(h(\theta)) = \int h(\theta) f(\theta) d\theta = \int \frac{h(\theta) f(\theta)}{g(\theta)} g(\theta) d\theta, \tag{12}$$

and approximate it with the Monte Carlo method

$$E(h(\theta)) \approx \frac{\sum_{l=1}^{L} h(\dot{\theta}_j) w(\dot{\theta}_j)}{\sum_{l=1}^{L} w(\dot{\theta}_j)}, \tag{13}$$

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 $p(y|x^{(i)}, D^{(\setminus i)}, M)$, we can calculate the Monte Carlo approximation of the expectation

$$E_y[g(y)|x^{(i)}, D^{(\setminus i)}, M] \approx \frac{1}{m} \sum_{j=1}^{m} g(\ddot{y}_j). \tag{14}$$

If $\ddot{\theta}_{ij}$ is a sample from $p(\theta|D^{(\setminus i)}, M)$ and we draw $\ddot{y}_j$ from $p(y|x^{(i)}, \ddot{\theta}_{ij}, M)$, then $\ddot{y}_j$ is a sample from $p(y|x^{(i)}, D^{(\setminus i)}, M)$. 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

$$w_j^{(i)} = \frac{p(\dot{\theta}_j|D^{(\setminus i)}, M)}{p(\dot{\theta}_j|D, M)} \propto \frac{1}{p(y^{(i)}|x^{(i)}, \dot{\theta}_j, D^{(\setminus i)}, M)}. \tag{15}$$

The reliability of the importance sampling can be estimated by examining the expected variability of the importance weights. For simple models this may be computed analytically (see, e.g., Peruggia, 1997), but if analytical solutions are inapplicable, we have to estimate this from the weights obtained. It is customary to examine the distribution of weights with various plots (see, e.g, Newton & Raftery, 1994; Gelman et al., 1995, chap. 10; Peruggia, 1997). We prefer plotting the cumulative normalized weights (see examples in section 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 based on an approximation of the variance of importance weights computed as

$$m_{\text{eff}}^{(i)} = 1/\sum_{j=1}^{m}(w_j^{(i)})^2, \tag{16}$$

where $w_j^{(i)}$ are normalized weights (Kong, Liu, & Wong, 1994; Liu, 2001, chap. 2.5.3). 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 4). Note that a small variance estimate of the obtained sample weights does not guarantee that importance sampling is giving the correct answer. On the other hand, the same 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 & 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, e.g., MacKay, 1998). 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 4.1 a simple case where IS-LOO-CV works well for flexible nonlinear models and in section 4.2 a case, which is more difficult and where IS-LOO-CV fails. In section 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 & 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 & 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 n is moderate, it might also be viable to use adaptive importance sampling methods (see, e.g., Zhang, 1996) for each i.

Importance-sampling might be improved by using model-specific importance link functions (MacEachern & Peruggia, 2000) or computationally more intensive adaptive importance sampling methods (see, e.g., Zhang, 1996). 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 2.2.

2.2 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 1 and 10)

$$p(y|x^{(i)}, D^{(\setminus s(i))}, M) = \int p(y|x^{(i)}, \theta, D^{(\setminus s(i))}, M)\, p(\theta|D^{(\setminus s(i))}, M) d\theta, \qquad (17)$$

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

$$\bar{u}_{\mathrm{CV}} = E_i[u(y^{(i)}, x^{(i)}, D^{(\setminus s(i))}, M)] \qquad (18)$$

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)

$$\begin{aligned}
\bar{u}_{\mathrm{tr}} &= E_i[u(y^{(i)}, x^{(i)}, D, M)] && (19)\\
\bar{u}_{\mathrm{cvtr}} &= E_j\big[E_i[u(y^{(i)}, x^{(i)}, D^{(\setminus s_j)}, M)]\big] \quad ; \quad j = 1, \ldots, k && (20)\\
\bar{u}_{\mathrm{CCV}} &= \bar{u}_{\mathrm{CV}} + \bar{u}_{\mathrm{tr}} - \bar{u}_{\mathrm{cvtr}}, && (21)
\end{aligned}$$

where $\bar{u}_{\mathrm{tr}}$ is the expected utility evaluated with the training data given the training data, that is, the training error or the expected utility computed with the posterior predictive densities (see section 3.2), and $\bar{u}_{\mathrm{cvtr}}$ is the average of the expected utilities evaluated with the training 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 the 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, unless n is very small. See discussion in the next section and some related discussion in (Burman, 1989).

We demonstrate in section 4.1 a simple case where the IS-LOO-CV and (bias corrected) k-fold-CV give equally good results and in section 4.2 a case, which is more difficult where the k-fold-CV works well and the IS-LOO-CV fails. In section 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, Chow, and Nolan (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. 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 are 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.

2.3 Distribution of the expected utility estimate

To assess the reliability of the expected utility estimate, 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 $p(y|x^{(i)}, D^{(\backslash s_j)}, M)$ are based on training data sets $D^{(\backslash s_j)}$, 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 4.1). If in doubt, this increase could be estimated as mentioned in section 4.1. See also comments in the next section.

If utilities u_i are summarized with the mean

$$\bar{u} = E_i[u_i], \tag{22}$$

a simple approximation would be to assume u_i's to have an approximately Gaussian distribution (described by the mean and the variance) and to compute the variance of the expected utility estimate as (see, e.g., Breiman, Friedman, Olshen, & Stone, 1984, chap. 11)

$$\mathrm{Var}[\bar{u}] = \mathrm{Var}_i[u_i]/n. \tag{23}$$

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. A variation of this, applicable in the k-fold-CV case, is that first the mean expected utility $\bar{u}_j$ for each of the k folds is computed and then the variance of the expected utility estimate is computed as (see, e.g., Dietterich, 1998)

$$\mathrm{Var}[\bar{u}] \approx \mathrm{Var}_j[\bar{u}_j]/k. \tag{24}$$

Here the distribution of $\bar{u}_j$'s tends to be closer to Gaussian (due to central limit theorem), but a drawback is that this estimator has a larger variance than the estimator of equation 23.

If the summary quantity is some other than mean (e.g., α-quantile) or the distribution of u_i's is 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 the 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, Carlin, Stern, & Rubin, 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 & Newton, 1992) for some important properties of the BB.

The assumption that 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 a 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 estimate $\bar{u}$, 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, Shao, & Ibrahim, 2000, chap. 7), 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, chap. 6; Chen et al., 2000, chap. 3)). However, if MCMC samples were highly dependent, we could use dependent weights in BB (Künsch, 1989, 1994).

2.4 Model comparison with expected utilities

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

$$\bar{u}_{M_1-M_2} = E_i[u_{M_1,i} - u_{M_2,i}]. \tag{25}$$

If the variability due to Monte Carlo integration is assumed negligible and a Gaussian approximation is used for the distributions of the expected utility estimates (equation 23 or equation 24), the probability $p(\bar{u}_{M_1-M_2} > 0)$ can be computed analytically.

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 utility estimates as

$$\dot{\bar{u}}_{(M_1-M_2),b} = \dot{\bar{u}}_{M_1,b} - \dot{\bar{u}}_{M_2,b}. \tag{26}$$

Then we can, for example, plot the distribution of $\bar{u}_{M_1-M_2}$ or compute the probability $p(\bar{u}_{M_1-M_2} > 0)$. Following simplicity postulate (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). Although possible overestimation of the variability due to training sets being slightly different (see the previous section) makes these comparisons slightly conservative, the error is small and in model choice, it is better to be conservative than too optimistic.

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 only a negligible amount of money.

The expected predictive densities have an important relation to Bayes factors, which are commonly used in Bayesian model comparison. If utility u is the pre-

dictive log-likelihood and (mean) expected utilities are computed by using cross-validation predictive densities then

$$\mathrm{PsBF}(M_1, M_2) \equiv \prod_{i=1}^{n} \frac{p(y^{(i)}|x^{(i)}, D^{(\setminus i)}, M_1)}{p(y^{(i)}|x^{(i)}, D^{(\setminus i)}, M_2)} = \exp(n\bar{u}_{M_1-M_2}), \tag{27}$$

where PsBF stands for pseudo-Bayes factor (Geisser & 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 4). Note that previously only point estimates for PsBF have been used, but with the proposed approach, it is possible to compute also the distribution of the PsBF estimate.

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. 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, Best, & Carlin, 1998, p. 3).

3 Relations to other predictive approaches

In this section, we discuss the relations of the cross-validation predictive densities to prior predictive densities and Bayes factors (section 3.1), and posterior predictive densities (section 3.2). See (Vehtari, 2001, chap. 3.3) for discussion of relations to other predictive densities, information criteria and the estimation of the effective number of parameters.

3.1 Prior predictive densities and Bayes factors

The marginal prior predictive densities (compare to equation 1)

$$p(y|x^{(i)}, M) = \int p(y|x^{(i)}, \theta, M)\, p(\theta|M) d\theta \tag{28}$$

are conditioned only on the prior, not on the data. The expected utilities computed with the marginal prior predictive densities would measure the goodness of the predictions without training samples used and could be used as an estimate of the lower (or upper, if a smaller value is better) limit for the expected utility.

The joint prior predictive densities $\int p(D|\theta, M)\, p(\theta|M) d\theta = p(D|M)$ are used to compute the Bayes factors (BF) $\mathrm{BF}(M_1, M_2) = p(D|M_1)/p(D|M_2)$ which are commonly used in Bayesian model comparison (Jeffreys, 1961; Kass & Raftery, 1995). 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. The prior sensitivity of the Bayes factor has been long known (Jeffreys, 1961) and is sometimes called "Lindley's Paradox" or "Bartlett's paradox" (see a nice review and historical comments in (Hill, 1982)).

If prior and likelihood are very different, normalized prior predictive densities may be very difficult to compute (Kass & Raftery, 1995). However, it may be possible to estimate unnormalized prior predictive likelihoods for large number of models relatively fast (see, e.g., Ntzoufras, 1999; Han & Carlin, 2001), so that prior predictive approach may be used to aid model selection as discussed by Vehtari (2001, chap. 4).

3.2 Posterior predictive densities

Posterior predictive densities are naturally used for new data (equation 1). When used for the training data, the expected utilities computed with the marginal 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 4). Comparison of the joint posterior predictive densities leads to the posterior Bayes factor (PoBF) (Aitkin, 1991). The posterior predictive densities should generally not be used either for assessing model performance, except as an estimate of the upper (or lower if smaller value is better) limit for the expected utility, or in model comparison as they favor overfitted models (see also discussion of paper (Aitkin, 1991)). Only if the effective number of parameters is relatively small, that is $p_{\text{eff}} \ll n$, the posterior predictive densities may be useful approximation to cross-validation predictive densities (Vehtari, 2001, chap. 3.3.4), 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), Gelman et al. (1995), and Gelman, Meng, and Stern (1996). 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. To avoid using the data twice, we have also used the cross-validation predictive densities for such analysis. This approach has also been used in some form by Gelfand et al. (1992), Gelfand (1996), and Draper (1995, 1996).

4 Illustrative examples

As illustrative examples, we use MLP networks and Gaussian processes with Markov Chain Monte Carlo sampling (Neal, 1996, 1997, 1999; Lampinen & Vehtari, 2001) in one toy problem: MacKay's robot arm, and two real world problems: concrete

quality estimation and forest scene classification. See (Vehtari & Lampinen, 2001, Appendix) for details of the models, priors and MCMC parameters. The MCMC sampling was done with the FBM[1] software and Matlab-code partly derived from the FBM and Netlab[2] toolbox. For convergence diagnostics, we used a visual inspection of trends, the potential scale reduction method (Gelman, 1996) and the Kolmogorov-Smirnov test (Robert & Casella, 1999). Importance weights for MLP and GP were computed as described in (Vehtari, 2001, Ch 3.2.2).

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. 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

$$y_1 = 2.0\cos(x_1) + 1.3\cos(x_1 + x_2) + e_1, \qquad y_2 = 2.0\sin(x_1) + 1.3\sin(x_1 + x_2) + e_2, \quad (29)$$

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) [3]. 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 an 8-hidden-unit MLP and a GP with normal (N) residual model.

Figure 1 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 2 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.

[1] `http://www.cs.toronto.edu/~radford/fbm.software.html`
[2] `http://www.ncrg.aston.ac.uk/netlab/`
[3] Available from `http://www.inference.phy.cam.ac.uk/mackay/Bayes_FAQ.html`

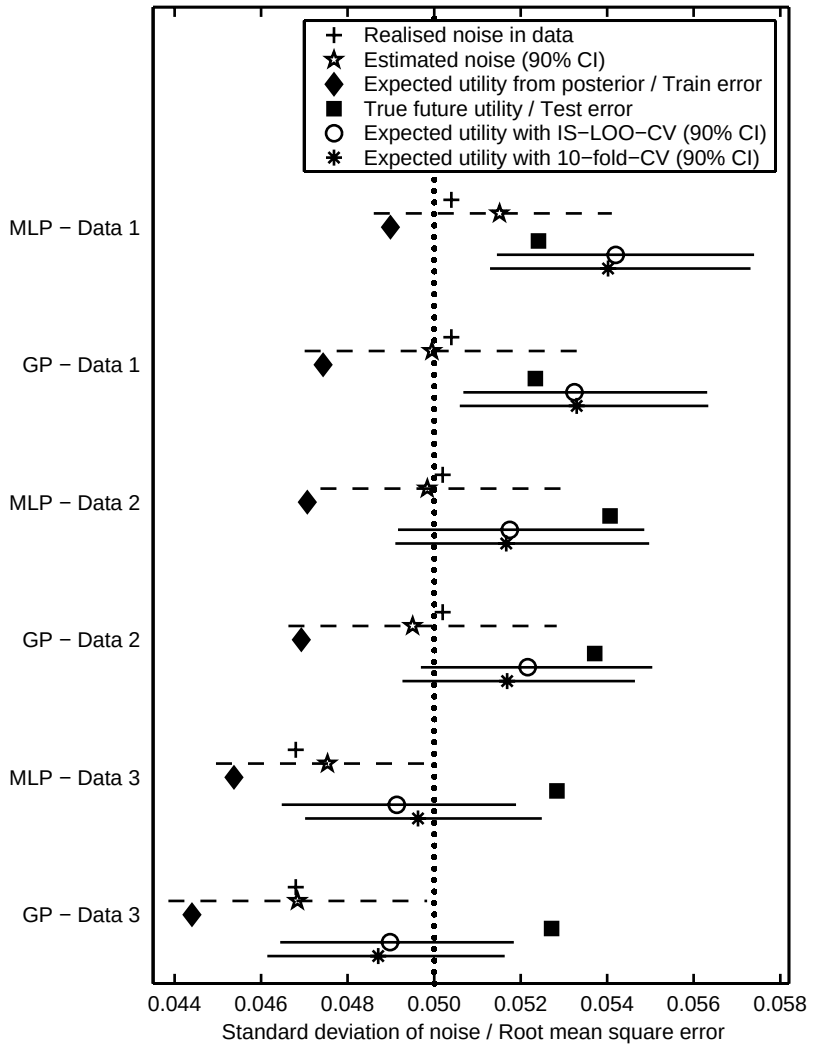

Figure 1: Robot arm: The expected utilities (root mean square errors) for MLPs and GPs. Results are shown for three different realizations of the data. The IS-LOO-CV and the 10-fold-CV give quite similar error estimates. Realized noise and estimated noise in each data set is also shown. Dotted vertical line shows the level of the theoretical noise.

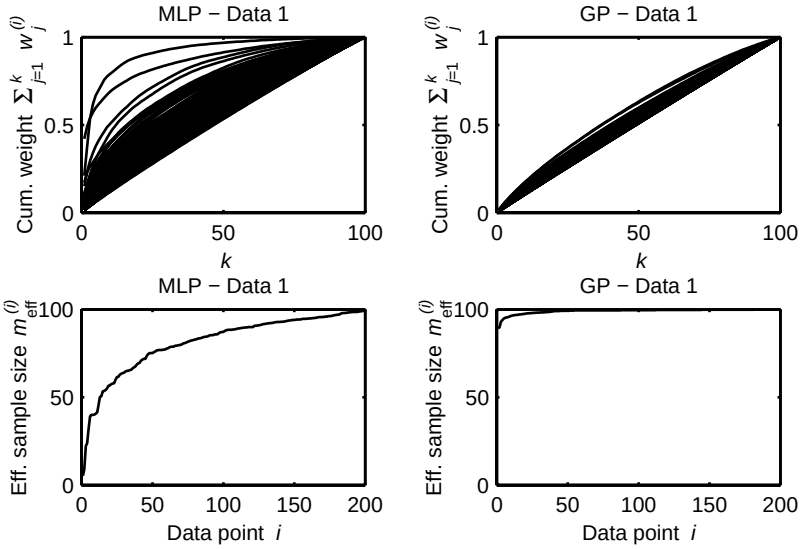

Figure 2: Robot arm: Top plots show the total cumulative mass assigned to the k largest importance weights versus k (one line for each data point i). The MLP has more mass attached to fewer weights. Bottom plots show the effective sample size of the importance sampling m_{eff}^{i} for each data point 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 unstable for the MLP, but probably stable for the GP.

In Figure 1 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 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 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

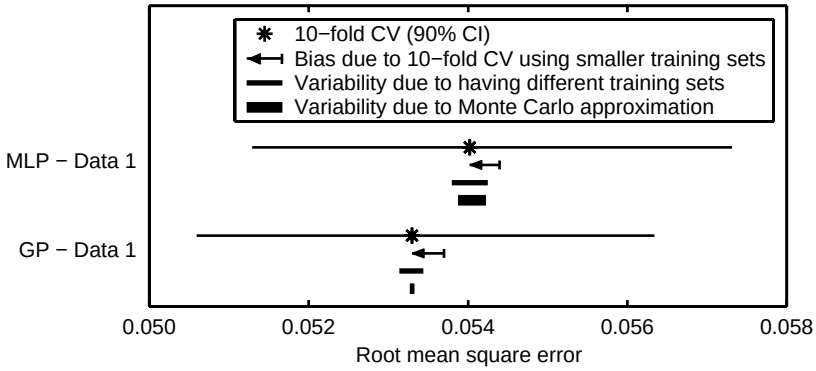

Figure 3: Robot arm: The different components that contribute to the uncertainty, and bias correction for the expected utility (root mean square errors) for MLP and GP. Results are shown for the data set 1. The variability due to having slightly different training sets in 10-fold-CV and the variability 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.

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 4.2 and 4.3 the bias correction had notable effect.

Figures 4 and 5 demonstrate the comparison of models using paired comparison of the expected utilities. Figure 4 shows the expected difference of root mean square errors and Figure 5 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 1). 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 2).

Figure 6 shows different components that contribute to the uncertainty in paired comparison 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 6 also shows that in this case, the bias in the 10-fold-CV is negligible.

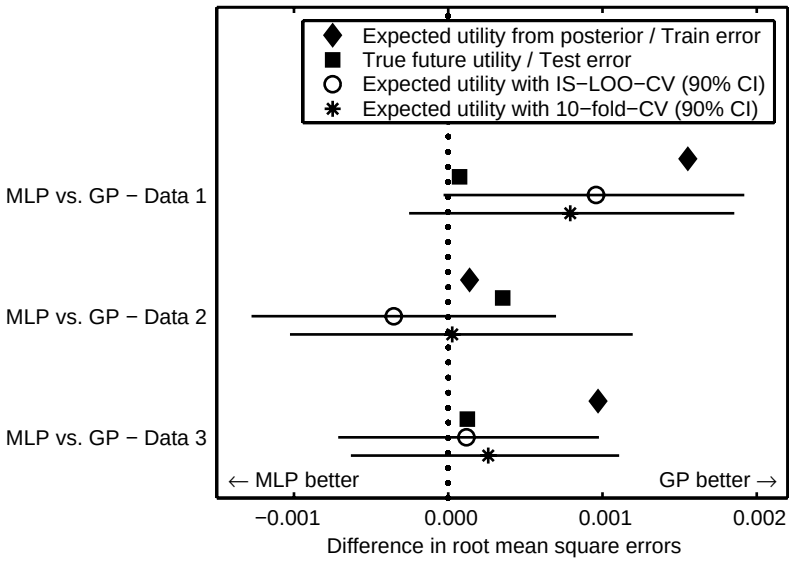

Figure 4: Robot arm: The expected difference of root mean square errors for MLP vs. GP. Results are shown for three different realizations of the data. The disagreement between the IS-LOO-CV and the 10-fold-CV shows slightly more clearly when comparing models than when estimating expected utilities (compare to Figure 1).

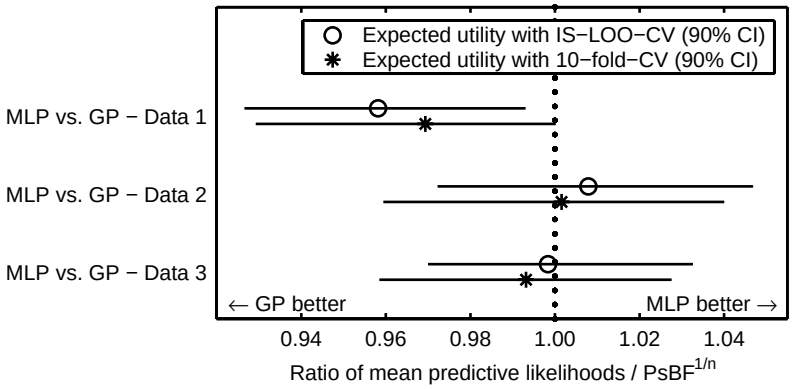

Figure 5: Robot arm: The expected ratio of mean predictive likelihoods (nth root of the pseudo-Bayes factors) for MLP vs. GP. Results are shown for three different realizations of the data. The disagreement between the IS-LOO-CV and the 10-fold-CV shows slightly more clearly when comparing models than when estimating expected utilities (compare to Figure 1).

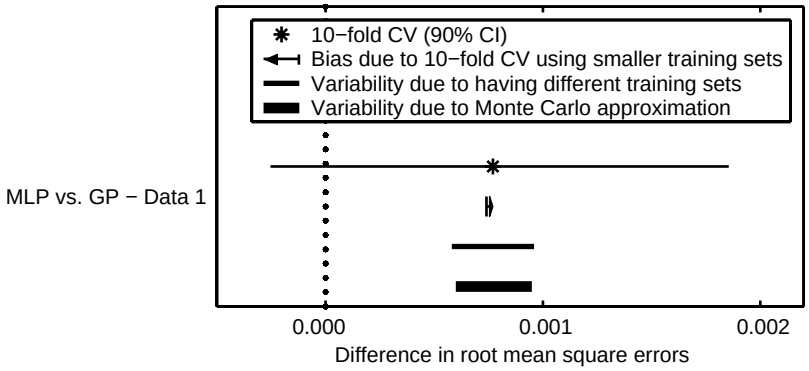

Figure 6: Robot arm: The different components that contribute to the uncertainty, and bias correction for the expected difference of the expected root mean square errors for MLP vs. GP. Results are shown for the data set 1. The variability due to having slightly different training sets in the 10-fold-CV 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.

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. 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), spread, slump and air-%, that 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 and 215 samples designed to cover the practical range of the variables, collected by the concrete manufacturing company. See the details of problem and the conclusions made by the concrete expert in (Järvenpää, 2001). 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

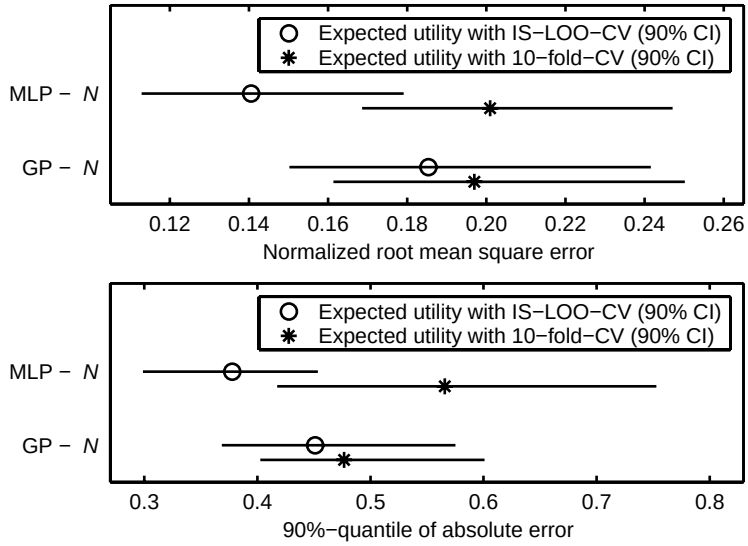

Figure 7: Concrete quality estimation: The expected utilities for MLP and GP with the Normal (N) residual model. The top plot shows the expected normalized root mean square errors and the bottom plot shows the expected 90%-quantiles of absolute errors.

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 7 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 8 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 9 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

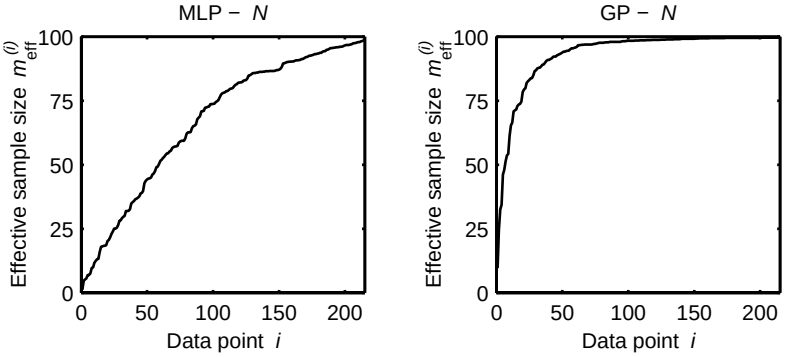

Figure 8: Concrete quality estimation: The effective sample sizes of the importance sampling $m_{\text{eff}}^{(i)}$ for each data point i (sorted in increasing order) for MLP and GP with the Normal (N) noise model. Both models have many data points with a small effective sample size, which implies that the IS-LOO-CV fails.

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 2(a), 2(b), and 2(c) 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 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.

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, that is assuming that in all future casts no additives or just one of the additives will be used (Figure 10).

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 MLP (Vehtari,

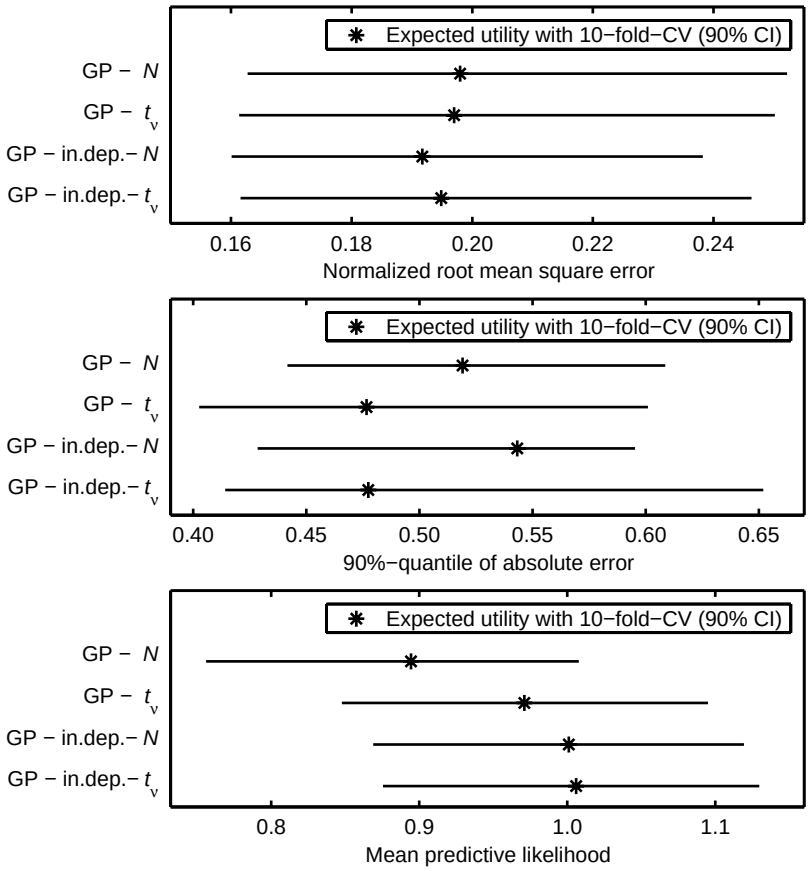

Figure 9: Concrete quality estimation: The expected utilities for GP models with Normal (N), Student's t_ν, input dependent Normal (in.dep.-N) and input dependent t_ν residual models. The top plot shows the expected normalized root mean square errors (smaller value is better), the middle plot shows the expected 90%-quantiles of absolute errors (smaller value is better) and the bottom plot shows the expected mean predictive likelihoods (larger value is better).

Table 1: Concrete quality estimation: Pairwise comparison of expected utilities of GP models with different residual models (see also Figure 9). The values in the matrices are probabilities that the model in the row is better than the model in the column.

		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	

(a) Normalized root mean square error

		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	

(b) 90%-quantile of absolute error

		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	

(c) Mean predictive likelihood

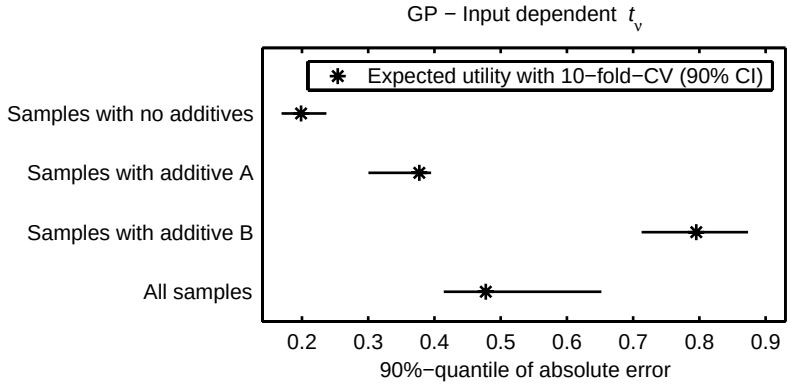

Figure 10: Concrete quality estimation: The expected utilities for the GP with the input dependent t_ν residual model. The plot shows the expected 90%-quantiles of absolute errors for samples with no additives, with additive A or B, and all samples.

Heikkonen, Lampinen, & Juujärvi, 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. We extracted a total of 84 potentially useful features: 48 Gabor filters (with different orientations and frequencies) that are generic features related to shape and texture, and 36 common statistical features (mean, variance and skewness with different window sizes). Fortyeight images were collected by 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.

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 expected 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 expected 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 Vehtari, 2001, chap. 4). As discussed in section 2.1 and demonstrated in section 4.2, leaving one point out can change posterior so much that importance sampling does not work. Leaving

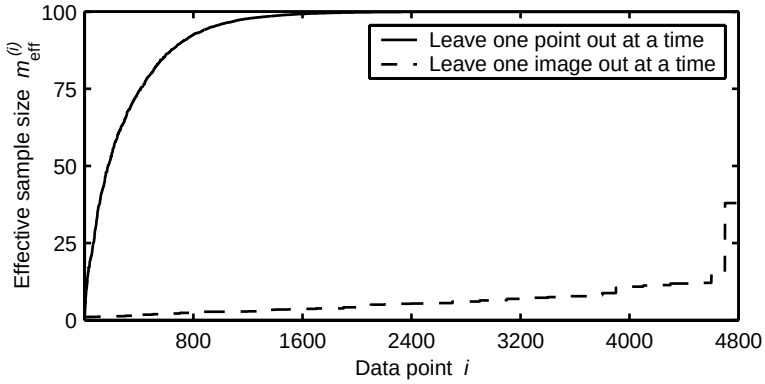

Figure 11: Forest scene classification: The effective sample sizes of the importance sampling $m_{\text{eff}}^{(i)}$ for each data point i (sorted in increasing order) for the 84-input logistic MLP. The effective sample sizes are calculated both for the leave-one-point-out (IS-LOO-CV) and the leave-one-image-out (IS-LOIO-CV) methods. In both cases there are many data points with a small effective sample size, which implies that importance sampling is unstable in this problem.

one image (100 data points) out will change posterior even more. Figure 11 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 12. The expected 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 11). 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.

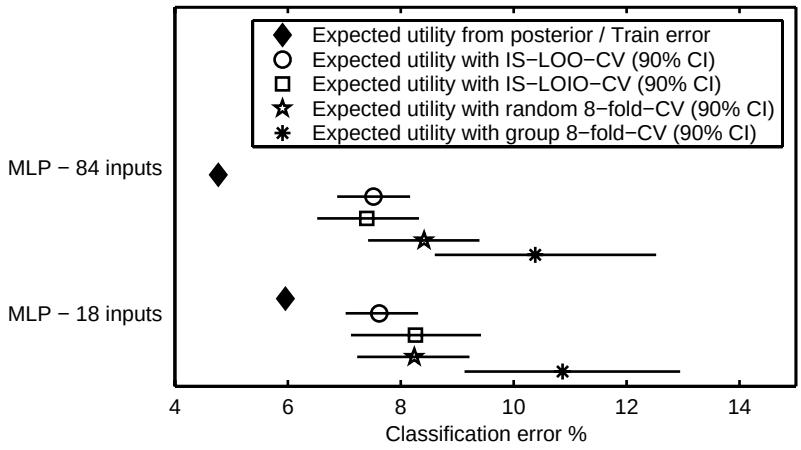

Figure 12: Forest scene classification: The expected utilities (classification errors) for the 84 and 18-input logistic MLPs.

5 Conclusions

The main goal of the paper was to give unified and formal presentation from Bayesian viewpoint how to compute the distribution of the expected utility estimate which can be used to describe, in terms of application field, how good the predictive ability of a Bayesian model is and how large is uncertainty in our estimate. The importance sampling leave-one-out predictive densities are a quick way to estimate the expected utilities and the approach is useful also 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 cross-validation predictive densities with the bias correction. Using k-fold-CV takes k times more time, but it is more reliable. In addition, if data points have certain dependencies, 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 utility estimates. With the proposed method, it is also easy to compute the probability that one model has better expected utility than another one.

Acknowledgements

This study was partly funded by TEKES Grant 40888/97 (Project *PROMISE, Applications of Probabilistic Modeling and Search*) and Graduate School in Electronics, Telecommunications and Automation (GETA). The authors would like to thank Dr. H. Järvenpää for providing her expertise into the concrete case study, and Prof. J.

Kaipio, Prof. H. Tirri, Prof. E. Arjas and anonymous reviewers for helpful comments.

References

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

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

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

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

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., & Nolan, D. (1994). A cross-validatory method for dependent data. *Biometrika*, 81(2), 351–358.

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

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

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

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

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

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

Geisser, S., & 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 W. R. Gilks, S. Richardson, & D. J. Spiegelhalter (eds.), *Markov Chain Monte Carlo in Practice*, (pp. 145–162). Chapman & Hall.

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

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

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

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

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

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

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

Good, I. J. (1952). Rational decisions. *Journal of the Royal Statistical Society B*, 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., & 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.

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

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

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.

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

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 B*, 56(1), 39.

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

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

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

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

MacEachern, S. N., & 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. (1998). Introduction to Monte Carlo methods. In M. I. Jordan (ed.), *Learning in Graphical Models*. Kluwer Academic Publishers.

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

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

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

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. Tech. Rep. 9702, Dept. of Statistics, University of Toronto.

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

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

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

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

Orr, M. J. L. (1996). Introduction to radial basis function networks [online]. Tech. rep., 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.

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

Robert, C. P., & 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.

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

Spiegelhalter, D. J., Best, N. G., & 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., Myles, J. P., Jones, D. R., & 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 B*, 36(2), 111–147.

Vehtari, A. (2001). *Bayesian Model Assessment and Selection Using Expected Utilities*. Dissertation for the degree of Doctor of Science in Technology, Helsinki University of Technology. Available also online at `http://lib.hut.fi/Diss/2001/isbn9512257653/`.

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

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

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

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.

Zhang, P. (1996). Nonparametric importance sampling. *Journal of the American Statistical Association*, 91(435), 1245–1253.