Helsinki University of Technology, Laboratory of Computational Engineering publications. Report B. ISSN 1457-1404

Bayesian Model Assessment and Comparison Using Cross-Validation Predictive Densities

Research report B27. ISBN 951-22-5630-4.

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

September 12, 2001

Abstract

We consider the problem of estimating the distribution of the expected utility of a Bayesian model. Our main goal is to describe in terms of application field how good the predictive ability of the model is and how large the uncertainty in our estimate is. We use cross-validation predictive densities to compute the expected utilities. We discuss and illustrate why in flexible non-linear models having many parameters, the quick importance sampling approximated leave-one-out cross-validation (IS-LOO-CV) proposed in (Gelfand et al., 1992) may not always work. 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 cross-validation (k-fold-CV). We also note that the k-fold-CV has to be used if data points have certain dependencies. As the k-fold-CV predictive densities are based on slightly smaller data sets than the full data set, we use a bias correction proposed in (Burman, 1989) to improve accuracy when computing the expected utilities. To assess the reliability of the estimated expected utilities, we propose a quick and generic approach based on the Bayesian bootstrap 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 future data distribution approximation. Moreover, it works better than the 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 comparison. With the proposed method, it is easy to compute the probability that one model has better expected utility than some other model. The proposed method can also be used to get samples from the distributions of the (prior), pseudo, posterior, partial, fractional and intrinsic Bayes factors by using corresponding predictive distributions and the predictive likelihood as a utility. As illustrative examples, we use MLP neural networks and Gaussian Processes (GP) with Markov Chain Monte Carlo sampling in one toy problem and two real world problems.

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

1 Introduction

Whatever way the model building and the selection has been made, the goodness of the final model should be somehow assessed in order to find out whether it is useful in a given problem. Even the best model selected from some collection of the 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 give good estimates of how reliable we believe our estimates to be. In prediction problems, it is natural to assess the predictive ability of the model by estimating the expected utilities. By using application specific utilities, the expected benefit or cost of using the model for predictions (e.g., measured by money) can be readily computed. In lack of application specific utilities, many general discrepancy and likelihood utilities can be used. The reliability of the estimated expected utility can be assessed by estimating the distribution of the expected utility.

Usually, utility is maximized, but we use the term more liberally. 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 which is most appealing for the application expert. It should be obvious in each case if smaller or larger value for the utility is better.

We use cross-validation predictive densities to compute the expected utilities. 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 a nice review of cross-validation and other predictive densities appears in (Gelfand, 1996). See also discussion in (Bernardo and Smith, 1994) how cross-validation approximates the formal Bayes procedure of computing the expected utilities. We review expected utilities and cross-validation predictive densities in sections 2.1 and 2.2.

For simple models, the cross-validation results can be computed quickly using analytical solutions. For more complex models where analytic solutions are not available, the model has to be built for each fold in cross-validation. A new idea in (Gelfand et al., 1992; Gelfand, 1996) was that instead of repeated model fitting, leave-one-out cross-validation could be approximated by using quick importance sampling (IS-LOO-CV) (section 2.3). However, this approximation may not always work in flexible non-linear models having many parameters. We discuss how the reliability of importance sampling can be estimated by examining the distribution of the importance weights and a heuristic measure of effective sample sizes. In case there is reason to suspect the reliability of the importance sampling, we suggest to use the predictive densities from the k-fold cross-validation (k-fold-CV) (section 2.4). As the k-fold cross-validation predictive densities are based on slightly smaller data sets than the full data set, we use a bias correction proposed in (Burman, 1989) to improve accuracy when computing the expected utilities. We also note and illustrate that the importance sampling approximation is unlikely to work if the data points have certain dependencies and several points have to be left out at a time (section 3.4).

To assess the reliability of the estimated expected utilities, we propose a quick and generic approach based on the Bayesian bootstrap (Rubin, 1981) for obtaining samples from the distributions of the expected utilities (section 2.5). The proposed approach can handle the variability due to Monte Carlo integration, bias correction estimation and future data distribution approximation. Moreover, it works better than the Gaussian approximation in the case of arbitrary summary quantities and non-Gaussian distributions.

Our main goal is model assessment but if there is a collection of models under consideration, by using the proposed method we can also easily compute the probability of one model having better expected utility than another one (section 2.6). The benefits of comparing the expected utilities are that the knowledge of how the model predictions are going to be used is taken into account and the expected utilities are also suitable in cases, where it is possible that none of the models is “true”.

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.

In the model assessment, the predictive likelihood as a utility would not be very descriptive for an application expert, but in model comparison, the predictive likelihood is a useful utility, as it measures how well the model predicts the predictive distribution. The expected predictive likelihood has important connections to Bayes factors which are commonly used in Bayesian model comparison (Kass and Raftery, 1995). Ratio of the expected CV-predictive likelihoods is also known as the pseudo-Bayes factor (Geisser and Eddy, 1979; Gelfand, 1996) and the (prior), posterior, partial, fractional and intrinsic Bayes factors (Kass and Raftery, 1995; Aitkin, 1991; O'Hagan, 1995; Berger and Pericchi, 1996) can be obtained by using corresponding predictive distributions (section 2.7). With the proposed method it is possible to obtain samples from the distributions of all these Bayes factors.

In section 2.8 we shortly discuss assumptions made on future data distribution in the approach described in this paper and in related approaches where the goal is to compare (not assess) the performance of methods instead of the models.

To illustrate the discussion we use MLP networks and Gaussian Processes (GP) with Markov Chain Monte Carlo (MCMC) sampling (Neal, 1996, 1999; Lampinen and Vehtari, 2001) in one toy problem and two real world problems (section 3).

We assume that the reader has basic knowledge of Bayesian methods (see, e.g., a short introduction in (Lampinen and Vehtari, 2001)). Knowledge of MCMC, MLP or GP methods is helpful but not necessary.

2 Methods

2.1 Expected utilities

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 $p(y|x^{(n+1)}, \theta, D, M) = p(y|x^{(n+1)}, \theta M)$. If the above integral is analytically intractable (e.g., in examples in section 3), the expectation of any function g can be estimated by using the Monte Carlo approximation

$$E_y[g(y)|x^{(n+1)}, D, M] \approx E_j[g(\dot{y}_j)] = \frac{1}{m} \sum_{j=1}^{m} g(\dot{y}_j), \tag{2}$$

where 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 drawn from $p(y|x^{(n+1)}, \dot{\theta}_j, D, M)$, then $\dot{y}_j$ is a sample from $p(y|x^{(n+1)}, D, M)$.

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 which 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 utility u

$$u_h = u(y^{(n+h)}, x^{(n+h)}, D, M). \tag{3}$$

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

$$\bar{u} = E_h[u_h] \tag{4}$$

or an α-quantile

$$\bar{u}_\alpha = Q_{\alpha,h}[u_h]. \tag{5}$$

We call all such summary quantities the expected utilities of the model. 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

$$u_h = (E_y[y|x^{(n+h)}, D, M] - y^{(n+h)})^2 \tag{6}$$

and the absolute error

$$u_h = \mathrm{abs}(E_y[y|x^{(n+h)}, D, M] - y^{(n+h)}) \tag{7}$$

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

$$u_h = p(y^{(n+h)}|x^{(n+h)}, D, M) \tag{8}$$

and it is especially useful in model comparison (see section 2.6).

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

$$\{p(y|x^{(i)}, D^{(\setminus i)}, M); i = 1, 2, \ldots, n\}, \tag{9}$$

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, for example,

$$\bar{u}_{\mathrm{loo}} = E_i[u(y^{(i)}, x^{(i)}, D^{(\setminus i)}, M)]. \tag{10}$$

If the future distribution of x is expected to be different from the distribution of the training data, this summary quantity could be changed to take this in account by weighting the observations appropriately (demonstrated in section 3.3).

The LOO-CV-predictive densities are computed with the equation (compare to Equation 1):

$$p(y|x^{(i)}, D^{(\setminus i)}, M) = \int p(y|x^{(i)}, \theta, D^{(\setminus i)}, M)\, p(\theta|D^{(\setminus i)}, M) d\theta. \tag{11}$$

For simple models, these LOO-CV-predictive densities may be computed quickly using analytical solutions, but models that are more complex usually require full model fitting for each n predictive distributions. 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 the next section or the k-fold-CV discussed in section 2.4 can be used to reduce the computational burden.

2.3 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 $h(\theta)$

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

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

$$E(h(\theta)) = \int \frac{h(\theta) f(\theta)}{g(\theta)} g(\theta) d\theta, \tag{13}$$

and approximate it with the Monte Carlo method

$$E(h(\theta)) \approx \frac{\sum_{l=1}^{L} h(\dot{\theta}_j) w(\dot{\theta}_j)}{\sum_{l=1}^{L} w(\dot{\theta}_j)}, \tag{14}$$

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

$$E_y[g(y)|x^{(i)}, D^{(\setminus i)}, M] \approx \frac{1}{m} \sum_{j=1}^{m} g(\ddot{y}_j). \tag{15}$$

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

$$w_j^{(i)} = \frac{p(\dot{\theta}_j|D^{(\setminus i)}, M)}{p(\dot{\theta}_j|D, M)} \propto \frac{1}{p(y^{(i)}|x^{(i)}, \dot{\theta}_j, D^{(\setminus i)}, M)}. \tag{16}$$

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 Bayesian linear model are given in (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; Peruggia, 1997). We prefer plotting the cumulative normalized weights (see section 3.2). 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), but when many different functions h are of potential interest, it is possible 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

$$m_{\text{eff}}^{(i)} = 1/\sum_{j=1}^{m} (w_j^{(i)})^2, \tag{17}$$

where w_{ij} are normalized weights (Kong et al., 1994; Liu and Chen, 1995). We propose to examine the distribution of the effective sample sizes by checking minimum and some quantiles and by plotting $m_{\text{eff}}^{(i)}$ in increasing order (see section 3). Note that this method cannot find out if the variance of the weights is infinite. However, as the importance sampling is unreliable also with a finite but large variance of weights, method can be in practice used to estimate the reliability of IS-LOO-CV. Also note that a small variance estimate of the obtained sample weights does not guarantee that importance sampling is giving correct answer, but on the other hand, similar problem applies to any variance or convergence diagnostics method based on finite samples of any non-direct Monte Carlo method (see, e.g., Neal, 1993; Robert and Casella, 1999).

Even in simple models like 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 (see 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, 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 3.2 a simple case where IS-LOO-CV works well for flexible nonlinear models and in section 3.3 a case that is more difficult where IS-LOO-CV fails. In section 3.4 we illustrate that the importance sampling does not work if data points have such dependencies that several points have to be left 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 the next section).

2.4 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 11):

$$p(y|x^{(i)}, D^{(\setminus s(i))}, M) = \int p(y|x^{(i)}, \theta, D^{(\setminus s(i))}, M)\, p(\theta|D^{(\setminus s(i))}, M) d\theta, \tag{18}$$

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 $k << n$, computational savings are considerable.

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

$$\bar{u}_{\text{cv}} = E_i[u(y^{(i)}, x^{(i)}, D^{(\setminus s(i))}, M)] \tag{19}$$

is biased. This bias has been usually ignored, maybe because k-fold-CV has been used mostly in model comparison, where biases cancel out if the models being compared have similar steepnesses of the learning curves. But in the case of different steepnesses of the learning curves and in model assessment, this bias should not be ignored. To get more accurate results, the bias corrected expected utility estimate $\bar{u}_{\text{ccv}}$ can be computed by using a less well

known first order bias correction (Burman, 1989)

$$\bar{u}_{\text{tr}} = E_i[u(y^{(i)}, x^{(i)}, D, M)] \tag{20}$$

$$\bar{u}_{\text{cvtr}} = E_j\left[E_i[u(y^{(i)}, x^{(i)}, D^{(\backslash s_j)}, M)]\right] \quad ; \quad j = 1, \ldots, k \tag{21}$$

$$\bar{u}_{\text{ccv}} = \bar{u}_{\text{cv}} + \bar{u}_{\text{tr}} - \bar{u}_{\text{cvtr}}, \tag{22}$$

where $\bar{u}_{\text{tr}}$ is the expected utility evaluated with the full data given full training data (i.e., training error) and $\bar{u}_{\text{cvtr}}$ 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 because with smaller k, (1) the k-fold-CV training data sets are more different from the full training data, (2) there are more ways to divide the training data randomly, but it is divided in just one way (3) 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 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 3.2 a simple case where the IS-LOO-CV and (bias corrected) k-fold-CV give equally good results and in section 3.3 a case that is more difficult where the k-fold-CV works well and the IS-LOO-CV fails. In section 3.4, we demonstrate a case where k-fold-CV works but IS-LOO-CV fails, as group dependencies in data require leaving several data points out at a time.

For the time series with finite range dependencies the k-fold-CV can be combined with the h-block-CV proposed in (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. 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.

2.5 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 approximating the future data distribution with a finite number of training data points. We are trying to estimate expected utilities given the training data D, but the cross-validation predictive densities $p(y|x^{(i)}, D^{(\backslash i)}, M)$ are based on training data sets $D^{(\backslash i)}$, which are each slightly different. This makes the u_i's slightly dependent in a way which 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 3.2). If in doubt, this increase could be estimated as mentioned in section 3.2. See also comments in next section.

If utilities u_i are summarized with mean

$$\bar{u} = E_i[u_i] \tag{23}$$

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

1984)

$$\mathrm{Var}[\bar{u}] = \mathrm{Var}_i[u_i]/n. \tag{24}$$

Of course, the distribution of u_i's is not necessarily Gaussian, but still this (or more robust variance estimate based on quantiles) is 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)

$$\mathrm{Var}[\bar{u}] \approx \mathrm{Var}_j[\bar{u}_j]/k. \tag{25}$$

Here $\bar{u}_j$ are distributed more Gaussianly, but drawback is that this estimator has much larger variance than the estimator of Equation 24.

If the summary quantity is other than mean (e.g., α-quantile) or the distribution of u_i's is considerably not Gaussian, above approximations may fail. Also 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 future data distribution approximation 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) and values of Z that are not observed have zero posterior probability. Sampling from the uniform 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, 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 future data distribution approximation). From obtained samples it is easy to compute, for example, credible intervals (CI), highest probability density intervals (see Chen et al., 2000), histograms or 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 $\bar{u}_i$.

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

2.6 Model comparison

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

$$\bar{u}_{M_1-M_2} = E_i[u_{M1,i} - u_{M2,i}]. \tag{26}$$

If the variability due to Monte Carlo integration is assumed to be negligible and Gaussian approximation is used for the distributions of the expected utilities (Equation 24 or Equation 25), p-value for the comparison can be computed by using paired t-test. 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 future data distribution approximations are dependent through i), we can get samples from the distribution of the difference of the expected utilities as

$$\dot{\bar{u}}_{M_1-M_2,b} = \dot{\bar{u}}_{M_1,b} - \dot{\bar{u}}_{M_2,b}. \tag{27}$$

Then we can, for example, plot the distribution of $\bar{u}_{M_1-M_2}$ or compute the probability $p(\bar{u}_{M_1-M_2} > 0)$. 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.

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

Expected predictive likelihood has important connection to Bayes factors which are commonly used for Bayesian model comparison. If utility u is the predictive log-likelihood and (mean) expected utilities are computed by using cross-validation predictive densities then

$$\mathrm{PsBF}(M_1, M_2) \equiv \prod_{i=1}^{n} \frac{p(y^{(i)}|x^{(i)}, D^{(\setminus i)}, M_1)}{p(y^{(i)}|x^{(i)}, D^{(\setminus i)}, M_2)} = \exp(n \times \bar{u}_{M_1-M_2}), \tag{28}$$

where PsBF stands for pseudo-Bayes factor (Geisser and Eddy, 1979; Gelfand, 1996). As we are interested in 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. Other types of Bayes factors are shortly discussed in next section.

As the proposed method 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).

2.7 Other predictive distributions

So far, we have concentrated on the CV-predictive distributions. In this section, we briefly review other alternatives (see also Gelfand, 1996; Gelfand and Dey, 1994). Note that method proposed in section 2.5 can be used to get samples from the distributions of the respective expected utilities or Bayes factors.

The prior predictive densities are mainly used to compute the prior predictive likelihoods $\prod_{i=1}^{n} p(y^{(i)}|x^{(i)} = p(D|M)$ which are used to obtain the (prior) Bayes factor $\mathrm{BF}(M_1, M_2) = p(D|M_1)/p(D|M_2)$ (Jeffreys, 1961; Kass and Raftery, 1995). The expected utilities computed by using the prior predictive densities would measure how good predictions do we get, if we have zero training samples (note that in order to have proper predictive distributions, the prior has to be proper). Clearly, the prior predictive densities should not be used for assessing model performance, except as an estimate of the lower (or upper, if smaller value is better) limit for the expected utility. In model comparison, BF specifically compares the goodness of the priors and so it is sensitive to changes in prior (Jeffreys, 1961; Kass and Raftery, 1995). Note that if prior and likelihood are very different, BF may be very difficult to compute (Kass and Raftery, 1995).

The posterior predictive distributions are naturally used for new data (Equation 1). When used for the training data, the expected utilities computed with the posterior predictive densities would measure how good predictions

do we get, if we use the same data for training and testing (i.e., future data samples would be exact replicates of the training data samples). This is equal to evaluating training error, which is well known to underestimate the generalization error of flexible models (see also examples in section 3). 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 be used 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 more overfitted models (see also discussion of paper Aitkin, 1991).

The partial predictive densities are based on old idea of dividing the data to two parts, that is, the training and the test set. Comparison of the partial predictive likelihhods $\prod_{i \in s} p(y^{(i)}|x^{(i)}, D^{(\setminus s)}, M) = p(D^{(s)}|D^{(\setminus s)}, M)$ 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, that is expectations of fractional likelihoods over fractional posteriors (Gilks, 1995). The use of fractional utilities makes it difficult to interpret the FBF in terms of normal predictive distributions and expected utilities.

The intrinsic Bayes factor (Berger and Pericchi, 1996) is computed by taking the arithmetic or geometric average of all such partial Bayes Factors which are computed by using all permutations of minimal subsets of training data that will make distribution $p(D^{(s)}|D^{(\setminus s)}, M)$ 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.

2.8 Results for new training data

In this section we shortly discuss about assumptions made on future data distribution in approach described in this paper and in related approaches (see, e.g., Rasmussen et al., 1996; Neal, 1998; Dietterich, 1998; Nadeau and Bengio, 1999, 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. We have conditioned our results on given realization of the training data D and we have assumed that the distribution of the future data for which we want to make predictions comes from the same distribution as the training data, that is, Ω (section 2.1). We estimate the variability due to approximative algorithm (Monte Carlo error) and variability due to approximating the distribution Ω with the training data (section 2.5).

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 1)

$$p(y|x^{(n+h)}, D_h^*, M) = \int p(y|x^{(n+h)}, \theta, D_h^*, M)\, p(\theta|D_h^*, M) d\theta. \tag{29}$$

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. In practice, the method comparison approaches have used various resampling, cross-validation and data splitting methods to produce proxies for D_h^* (see, e.g., Dietterich, 1998; Nadeau and Bengio, 1999, and references therein). The reuse of training samples is more difficult than in the approach described in this paper 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 need to assess the performance of the model.

When solving a real problem, is there 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 thrown away. 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. We could also give rough estimate what the performance of the model would be with additional training data before getting them, but it may be difficult because of the difficulties in estimating the shape of the learning curve.

We might want to throw the old training data away, if we assume that the future data comes from some other distribution Ω^+ and we would also get new training data D^+ from that distribution. Uncertainty due to getting new training data could be estimated 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 (Wolpert, 1996a,b; Lemm, 1996, 1999). 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 new domain (it is probable easier to estimate the relative performance of two models than the performance of 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 Illustrative examples

As illustrative examples, we use MLP networks and Gaussian Processes (GP) with Markov Chain Monte Carlo sampling (Neal, 1996, 1997, 1999; Lampinen and Vehtari, 2001) in one toy problem (MacKay's robot arm) and two real world problems (concrete quality estimation and forest scene classification).

3.1 MLP and GP models

Both MLP and GP are flexible nonlinear models, where the available number of parameters p in model may be near or even greater than the number of data samples n and also the effective number of parameters p_{eff} (MacKay, 1992; Spiegelhalter et al., 1998) is usually large compared to n.

We used one hidden layer MLP with tanh hidden units, which in matrix format can be written as

$$f(x, \theta_w) = b_2 + w_2 \tanh(b_1 + w_1 x)\,. \tag{30}$$

The 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. We used Gaussian priors on weights and the Automatic Relevance Determination (ARD) prior on input weights. In regression problems we used probability model with additive error

$$y = f(x; \theta_w) + e, \tag{31}$$

where the random variable e is the model residual. In two class classification problem, we used logistic transfor-

mation to compute the probability that a binary-valued target, y, has value 1

$$p(y = 1|x, \theta_w, M) = [1 + \exp(-f(x, \theta_w))]^{-1}. \tag{32}$$

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 as the term $p(y^{(i)}|x^{(i)}, \dot{\theta}_j, D^{(\setminus i)}, M)$ in Equation 16 simplifies to

$$p(y^{(i)}|x^{(i)}, \dot{\theta}_j, D^{(\setminus i)}, M) = p(y^{(i)}|x^{(i)}, \dot{\theta}_j, M). \tag{33}$$

Gaussian Process model is a non-parametric regression method, with priors imposed directly on the covariance function of the resulting approximation. Given the training inputs $x^{(1)}, \ldots, x^{(n)}$ 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 $y^{(1)}, \ldots, y^{(n)}, y^{n+1}$. 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

$$E_y[y|x^{(n+1)}, \theta, D] = k^{\mathrm{T}} C^{-1} y^{(1,\ldots,n)} \tag{34}$$

$$\mathrm{Var}_y[y|x^{(n+1)}, \theta, D] = V - k^{\mathrm{T}} C^{-1} k, \tag{35}$$

where C is the n by n covariance matrix of the observed targets, $y^{(1,\ldots,n)}$ 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

$$C_{ij} = \eta^2 \exp\left(-\sum_{u=1}^{p} \rho_u^2 (x_u^{(i)} - x_u^{(j)})^2\right) + \delta_{ij} J^2 + \delta_{ij} \sigma_e^2. \tag{36}$$

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. We use Inverse-Gamma prior on η^2 and hierarchical Inverse-Gamma prior (producing ARD like prior) on ρ_u. The second term is the jitter term, where $\delta_{ij} = 1$ when $i = j$. It is used to improve matrix computations by adding constant term to residual model. The third term is the residual model. For GP model, the predictions are dependent of both the parameters of the covariance function and the training data which means that the simplification of Equation 33 can not be used. However, the term $p(y^{(i)}|x^{(i)}, \dot{\theta}_j, D^{(\setminus i)}, M)$ in Equation 16 can be computed quickly using LOO results for GP with fixed hyperparameters from (Sundararajan and Keerthi, 2001)

$$\log p(y^{(i)}|x^{(i)}, \dot{\theta}_j, D^{(\setminus i)}, M) = \frac{1}{2}\log(2\pi) - \frac{1}{2}\log \bar{c}_{ii} + \frac{1}{2}\frac{q_i^2}{\bar{c}_{ii}}, \tag{37}$$

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 by using the parameters $\dot{\theta}_j$. Additionally it is useful to compute the leave-one-out predictions with given hyperparameters

$$E_y[y|x^{(i)}, \dot{\theta}_j, D^{(\setminus i)}, M] = y^{(i)} - \frac{q_i}{\bar{c}_{ii}} \tag{38}$$

$$\mathrm{Var}_y[y|x^{(i)}, \dot{\theta}_j, D^{(\setminus i)}, M] = \frac{1}{\bar{c}_{ii}}. \tag{39}$$

In the MCMC framework for MLP introduced in (Neal, 1996), sampling of the weights is done using the hybrid Monte Carlo (HMC) algorithm (Duane et al., 1987) and sampling of the hyperparameters (i.e., all the other parameters than weights) is done using the Gibbs sampling (Geman and Geman, 1984). In the case of the GP, parameters of the covariance function are sampled using the HMC, and per-case variances are sampled using the Gibbs sampling. 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 and Casella, 1999).

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

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

3.2 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. Very simple "robot arm" toy-problem (first used in MacKay, 1992) was selected, so that the complexity of the problem would not hide the main points we wanted to illustrate. Additionally we wanted to demonstrate uncertainties in this problem as it has been used in many papers without reporting uncertainty in error estimates and also it seems probable that different sets of test data has been used in some papers which has lead to overconfident conclusions.

The task is to learn the mapping from joint angles to position for 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

$$y_1 = 2.0\cos(x_1) + 1.3\cos(x_1 + x_2) + e_1 \tag{40}$$

$$y_2 = 2.0\sin(x_1) + 1.3\sin(x_1 + x_2) + e_2, \tag{41}$$

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 in (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 8 hidden unit MLP with 47 parameters and GP model with 4 parameters (for GP model, training samples could be considered as parameters too). In both cases, we used Normal (N) residual model. One hundred samples were drawn from the full posterior.

Figure 1 shows results from the MacKay's Robot Arm problem where the utility is root mean square error. IS-LOO-CV and 10-fold-CV give quite similar error estimates. Figure 2 shows that the importance sampling works probably very well for GP but it might produce wrong results for MLP. Although importance sampling weights for MLP are not very good, 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 get swamped by the uncertainty from not knowing the noise variance.

In Figure 1, realized, estimated and theoretical noise in each data set is also 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 of the uncertainty in the estimate of the expected utility. 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 estimate of the variability due to having slightly different training sets in 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 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 10-fold-CV is quite small. As the true function was known, we also computed estimate for the bias using the test data. For GP, the bias correction and the "true" bias were the same with about 2% accuracy. For MLP there was much more variation, but still all 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 3.3 and 3.4 the bias correction had slight but practically notable effect.

[3] Available from `http://wol.ra.phy.cam.ac.uk/mackay/Bayes_FAQ.html`

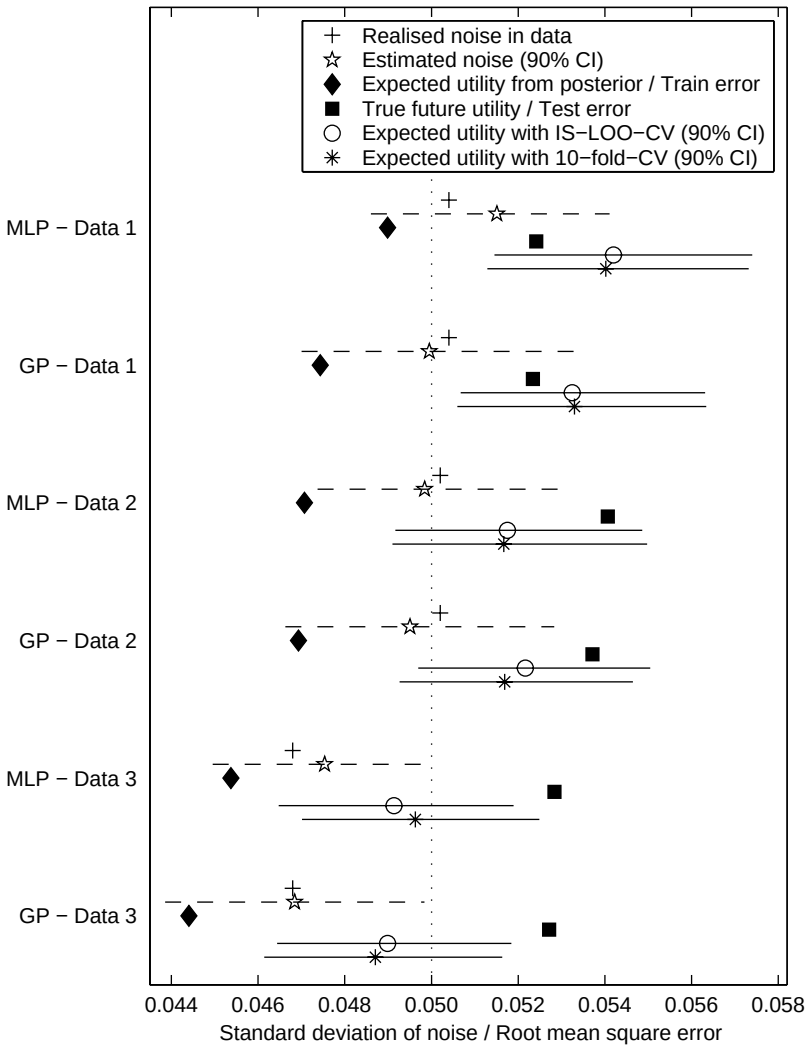

Figure 1: Robot arm example: The expected utilities (root mean square errors) for MLPs and GPs. Results are shown for three different realizations of the data. IS-LOO-CV and 10-fold-CV give quite similar error estimates. Realized noise and estimated noise in each data set is also shown. Dotted vertical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

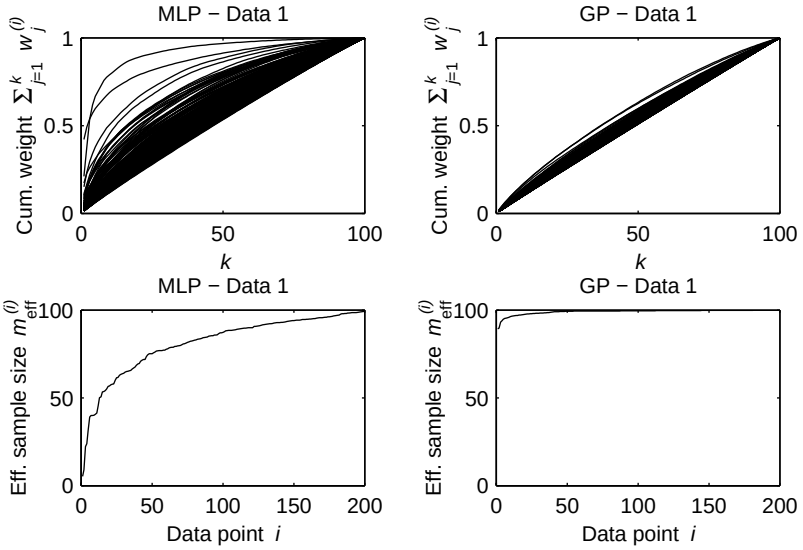

Figure 2: Robot arm example: Two plot types were used to visualize the reliability of the importance sampling. Top plots show the total cumulative mass assigned to the k largest importance weights versus k (one line for each data point i). MLP has more mass attached to fewer weights. Bottom plots show 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, IS-LOO-CV may be unreliable for the MLP, but probably works well for the GP.

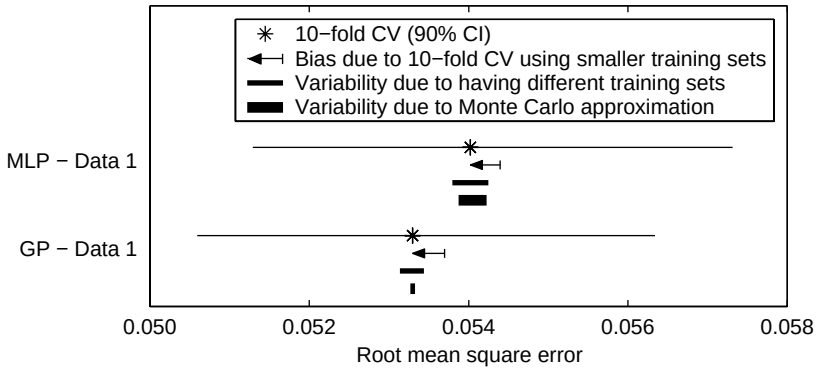

Figure 3: Robot arm example: The components of the uncertainty and bias correction for the expected utility (root mean square errors) for MLP and GP. Results are shown for the data set 1. The variability due to having slightly different training sets in 10-fold-CV and the variability due to the Monte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

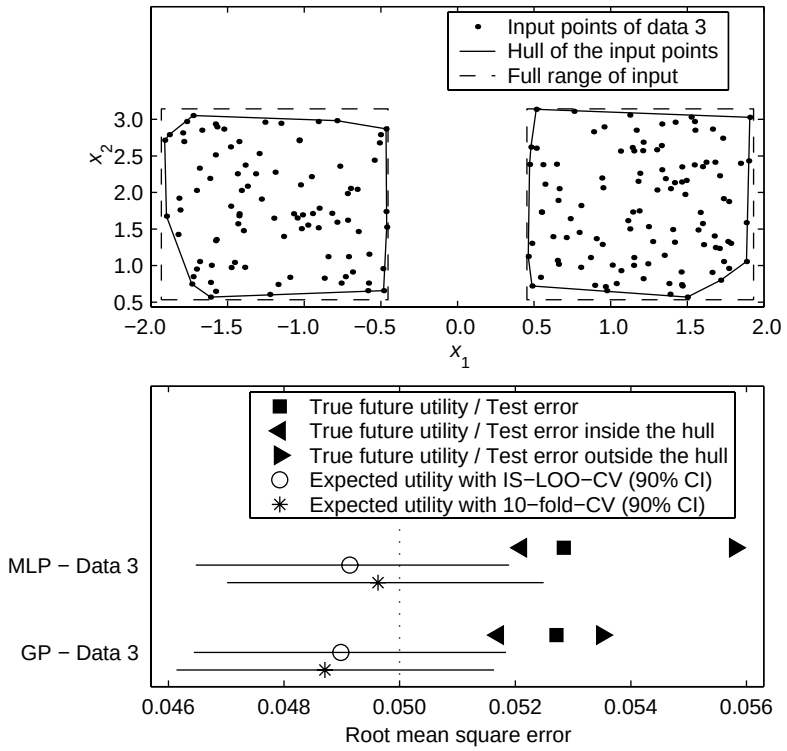

Figure 4: Robot arm example: The upper plot shows input points of data set 3, with the full range (broken line) and with the realized range approximated by two convex hulls (solid line). The lower plot shows how the true future utility (test error) inside the hull coincides better with credible interval for estimated expected utility.

Figure 4 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 be 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 5 and 6 demonstrate the comparison of models using paired comparison of the distributions of the expected utilities. Figure 5 shows the expected difference of root mean square errors and Figure 6 shows the expected ratio of mean predictive likelihoods (nth root of the pseudo-Bayes factors). IS-LOO-CV and 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). Disagreement between IS-LOO-CV and 10-fold-CV might be caused by bad importance weights of IS-LOO-CV for the MLPs (see Figure 2).

Figure 7 shows different components of uncertainty in paired comparison of the distributions of the expected utilities. The variability due to having slightly different training sets in 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 7 also shows that in this case, the bias in 10-fold-CV is negligible.

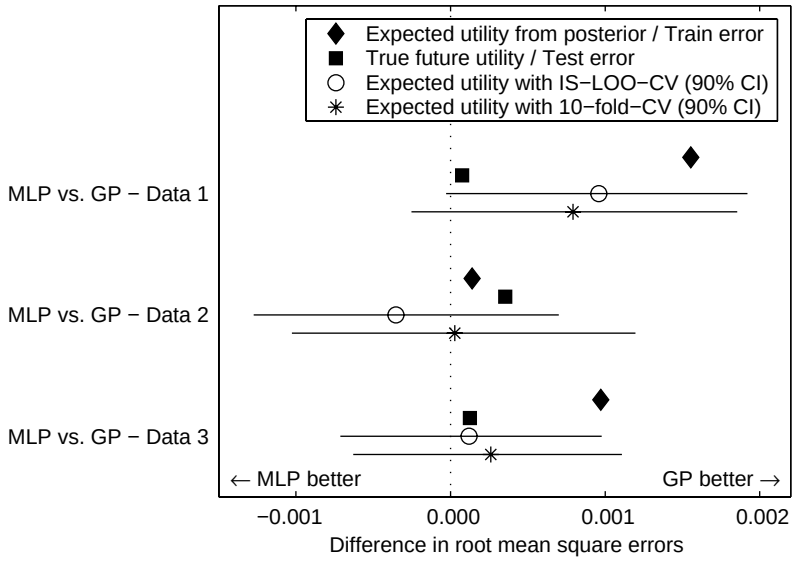

Figure 5: Robot arm example: The expected difference of root mean square errors for MLP vs. GP. Results are shown for three different realizations of the data. Disagreement between IS-LOO-CV and 10-fold-CV shows slightly more clearly when comparing models than when estimating expected utilities (compare to Figure 1). Figure 2 shows reason to suspect the reliability of IS-LOO-CV.

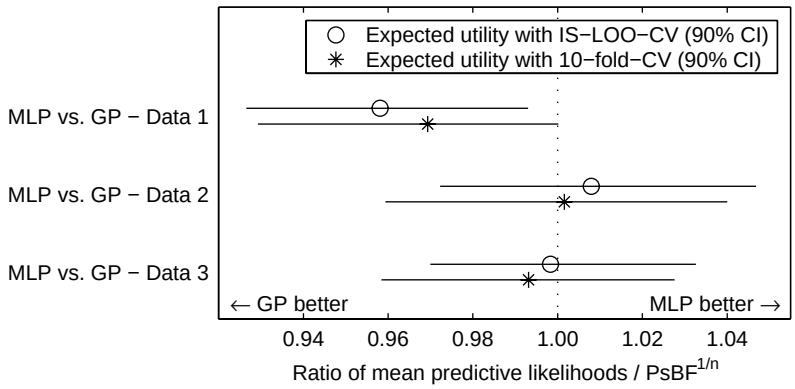

Figure 6: Robot arm example: Expected ratio of mean predictive likelihoods (nth root of the pseudo-Bayes Factors) for MLP vs. GP. Results are shown for three different realizations of the data. Disagreement between IS-LOO-CV and 10-fold-CV shows slightly more clearly when comparing models than when estimating expected utilities (compare to Figure 1). Figure 2 shows reason to suspect the reliability of IS-LOO-CV.

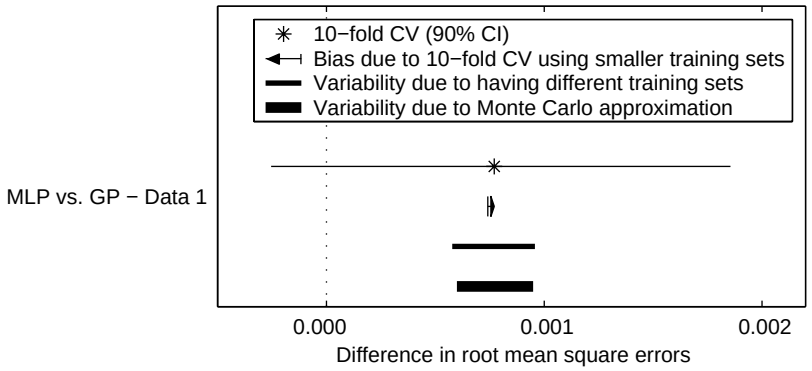

Figure 7: Robot arm example: The components of the uncertainty and bias correction for the expected difference of the expected root mean square errors for MLP vs. GP. Results are shown for the data set 1. The variability due to having slightly different training sets in 10-fold-CV and the variability 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 cancel out and combined bias correction is negligible.

3.3 Real world problem 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). It was very important to be able to describe the quality of the model in terms of the concrete expert instead of statistical jargon. It was also important to give good estimates of how reliable we believe our estimates were. In the following, we report the results for one variable, *air-%*, which measures the volume percentage of air in the concrete.

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 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. One hundred samples were drawn from the full posterior.

Figure 8 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 easily understandable. IS-LOO-CV gives much lower estimates for MLP and somewhat lower estimates for GP than 10-fold-CV. Figure 9 shows that IS-LOO-CV for both MLP and GP has many data points with small (or very small) effective sample size which implies that IS-LOO-CV cannot be used in this problem.

Figure 10 shows the expected normalized root mean square errors, the expected 90%-quantiles of absolute

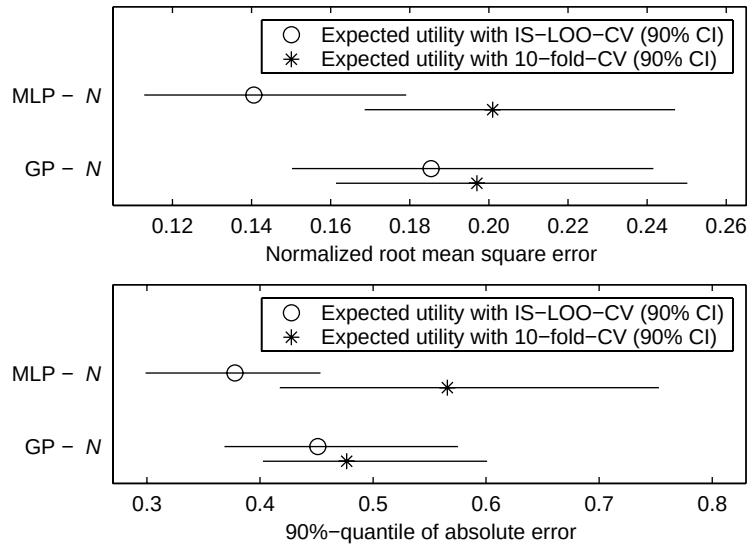

Figure 8: Concrete quality estimation example: The expected utilities for MLP and GP with the Normal (N) residual model. The top plot shows the expected normalized root mean square errors and the bottom plot shows the expected 90%-quantiles of absolute errors. IS-LOO-CV gives much lower estimates for MLP and somewhat lower estimates for GP than 10-fold-CV. Figure 9 shows reason to distrust IS-LOO-CV.

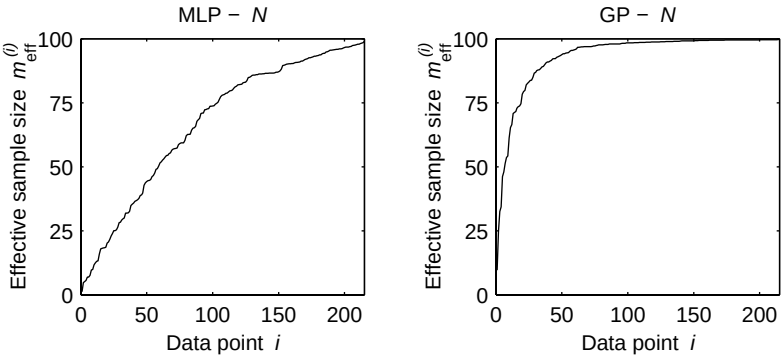

Figure 9: Concrete quality estimation example: The effective sample sizes of the importance sampling $m_{\text{eff}}^{(i)}$ for each data point i (sorted in increasing order) for MLP and GP with the Normal (N) noise model. Both models have many data points with small effective sample size which implies that IS-LOO-CV cannot be trusted.

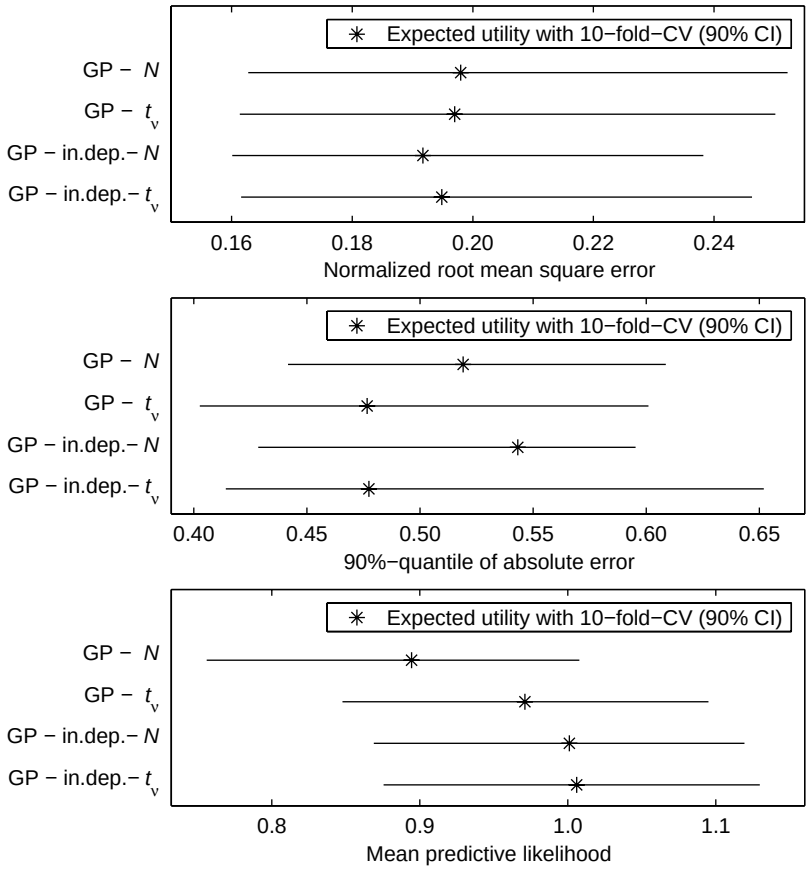

Figure 10: The expected utilities for GP models with Normal (N), Student's t_ν, input dependent Normal (in.dep.-N) and input dependent t_ν residual models. The top plot shows the expected normalized root mean square errors (smaller value is better), the middle plot shows the expected 90%-quantiles of 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 is larger differences if mean predictive likelihood is used instead (it is harder to guess the distribution of the prediction). See Tables 1, 2, and 3 for the pairwise comparisons of the residual models.

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 is larger differences if mean predictive likelihood is used instead (it is harder to guess the whole distribution of the guess). The bias corrections are not shown but they were about 3-5% of the median values, i.e, they have notable effect in model assessment. The biases were similar in different models, so they more or less cancel out in model comparison.

Tables 1, 2, and 3 show the results for the pairwise comparisons of the residual models. In this case, the uncertainties in 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 fear of choosing a bad model.With the mean predictive likelihood utility, there is more difference as it measures the accuracy in tails better. If 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 additives have strong influence on the quality of concrete, it was useful to report also the expected

Table 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 10). 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 10) 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 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 10). 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 10) 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: Concrete quality estimation example: Pairwise comparison of GP models with different residual models using mean predictive likelihood as utility (see also Figure 10). 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	

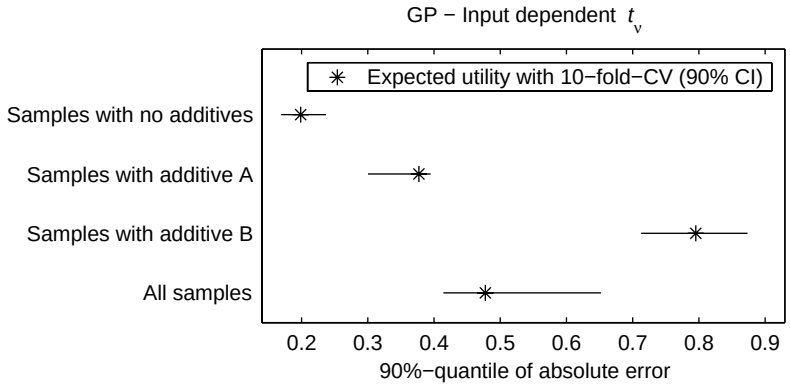

Figure 11: Concrete quality estimation example: The expected utilities depending on the additives used for GP with the input dependent t_ν residual model. Knowing that additives have strong influence on the quality of concrete, it was useful to report also the expected utility separately for the samples with different additives. Plot shows the expected 90%-quantiles of absolute errors for samples with no additives, with additive A or B, and all samples.

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 11 shows for 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.

3.4 Real world problem 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 reasonable results.

The case problem is the classification of forest scenes with MLP (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. 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.

We tested two 20 hidden unit MLPs with logistic likelihood model. First MLP used all 84 inputs and second MLP used a reduced set of 18 inputs selected using Reversible Jump MCMC (RJMCMC) method (Green, 1995; Vehtari and Lampinen, 2001). One hundred samples were drawn from the full posterior.

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 k-fold-CV, training and test sets (may) have data points from the same image, which would lead to a too optimistic estimates of the predictive utility. To get a realistic estimate of the predictive utility for new unseen images, training data set has to be divided by images.

As discussed in section 2.3 and demonstrated in section 3.3 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 12 shows the effective sample sizes of the importance sampling for the 84 input MLP for IS-LOO-CV and IS-LOIO-CV (leave-one-image-out) (for the 18 input MLP the plot was similar).

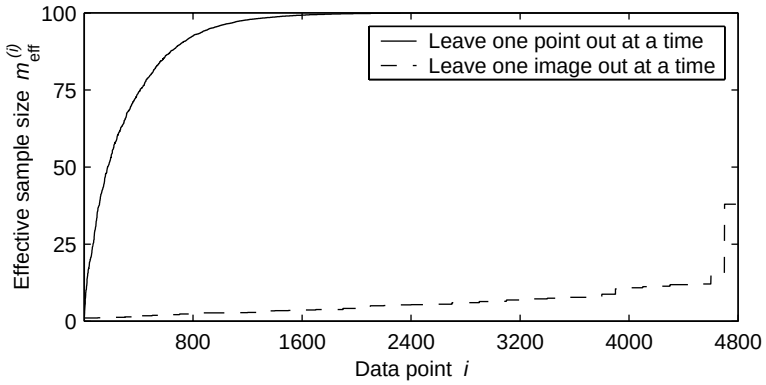

Figure 12: Forest scene classification example: The effective sample sizes of the importance sampling $m_{\text{eff}}^{(i)}$ for each data point i (sorted in increasing order) for 84 input logistic MLP. The effective sample sizes are calculated both for leave-one-point-out (IS-LOO-CV) and leave-one-image-out (IS-LOIO-CV). As data points from one image are dependent, cross-validation should be done by leaving one (or many) image(s) out at a time, but then posterior distribution changes too much to get reasonable importance weights. In this case, neither IS-LOO-CV nor IS-LOIO-CV can be trusted.

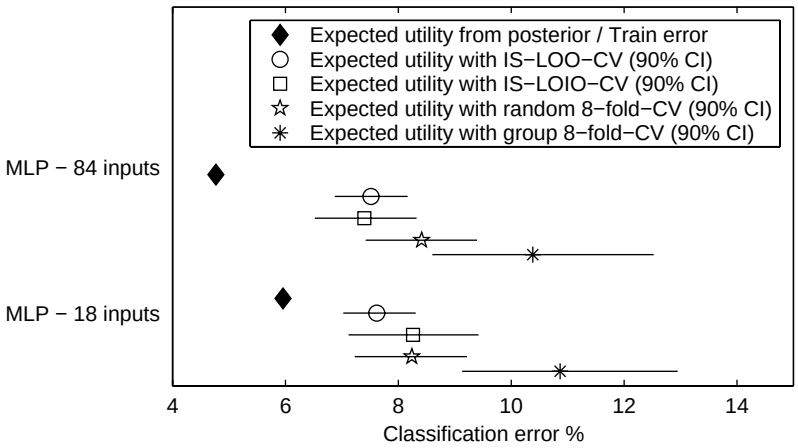

Figure 13: Forest scene classification example: The expected utilities (classification errors) for 84 and 18 input logistic MLPs. IS-LOO-CV gives too low estimate because data points from one image are dependent (and also because of somewhat bad importance weights) and IS-LOIO-CV gives too low estimate because of bad importance weights when leaving one image out at time (see Figure 12). 8-fold-cv with random data division gives too low estimate because data points from one image are dependent. In group 8-fold-CV, the data division was made by handling all the data points from one image as one indivisible group.

The expected classification errors for 84 and 18 input MLPs are shown in Figure 13. The predictive utility computed from the posterior predictive distribution (train error) gives too low estimates. IS-LOO-CV and 8-fold-CV with random data division give too low estimates because data points from one image are dependent. IS-LOO-CV also suffers from somewhat bad importance weights and IS-LOIO-CV suffers from very bad importance weights (see also Figure 12). In 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 84 and 18 input MLPs about 9% and 3% of the median values respectively. Note that more complex model had naturally steeper learning curve and correspondingly larger bias correction. In this case, biases did not cancel totally out in model comparison.

Pairwise comparison computed from group 8-fold-CV predictive distributions gives probability 0.86 that 84 input model has lower expected classification error than 18 input model. We still might use the smaller model for classification, as it would be not much worse, but slightly faster.

4 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 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 IS-LOO-CV predictive densities are a quick way to estimate the expected utilities and the approach is useful also in some cases with flexible non-linear 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 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 utilities. 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 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.

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.

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

Breiman, L., Friedman, J., Olshen, R., and 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., and Nolan, D. (1994). A cross-validatory method for dependent data. *Biometrika*, 81(2):351–358.

Carlin, B. P. and 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., and Ibrahim, J. Q. (2000). *Monte Carlo Methods in Bayesian Computation*. Springer.

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

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

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*, pages 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 B*, 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*, pages 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*, pages 131–144. Chapman & Hall.

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

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.

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

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

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

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.

Kass, R. E. and 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). Discussion: Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society B*, 56(1):39.

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.

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

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. 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. (1998). 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.

Nadeau, C. and Bengio, Y. (1999). Inference for the generalization error. Technical Report 99s-25, CIRANO, Montreal.

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. (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*, pages 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*, pages 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 B*, 56(1):3–48.

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

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.

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.

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`.

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.

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., and Carlin, B. P. (1998). Bayesian deviance, the effective number of parameters, and the comparison of arbitrarily complex models. Technical Report 98-009, 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 B*, 36(2):111–147.

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

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*, pages 66–73. SPIE.

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

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.