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

Bayesian Input Variable Selection Using Posterior Probabilities and Expected Utilities

Research report B31. ISBN 951-22-6229-0.

Aki Vehtari and Jouko Lampinen
Laboratory of Computational Engineering
Helsinki University of Technology
P.O.Box 9203, FIN-02015, HUT, Finland
{Aki.Vehtari,Jouko.Lampinen}@hut.fi

May 28, 2002
Revised December 20, 2002

Abstract

We consider the input variable selection in complex Bayesian hierarchical models. Our goal is to find a model with the smallest number of input variables having statistically or practically at least the same expected utility as the full model with all the available inputs. A good estimate for the expected utility can be computed using cross-validation predictive densities. In the case of input selection and a large number of input combinations, the computation of the cross-validation predictive densities for each model easily becomes computationally prohibitive. We propose to use the posterior probabilities obtained via variable dimension MCMC methods to find out potentially useful input combinations, for which the final model choice and assessment is done using the expected utilities. Variable dimension MCMC methods visit the models according to their posterior probabilities. As models with negligible probability are probably not visited in a finite time, the computational savings can be considerable compared to going through all possible models. If there is problem of obtaining enough samples in reasonable time to estimate the probabilities of the models well, we propose to use the marginal posterior probabilities of the inputs to estimate their relevance. As illustrative examples we use MLP neural networks and Gaussian processes in one toy problem and in two challenging real world problems. Results show that using posterior probability estimates computed with variable dimension MCMC helps finding useful models. Furthermore, benefits of using expected utilities for input variable selection are that it is less sensitive to prior choices and it provides useful model assessment.

Keywords: Bayesian model choice; input variable selection; expected utility; cross-validation; variable dimension Markov chain Monte Carlo; MLP neural networks; Gaussian processes; Automatic Relevance Determination

1 Introduction

In practical problems, it is often possible to measure many variables, but it is not necessarily known which of them are relevant and required to solve the problem. In Bayesian hierarchical models, it is usually feasible to use large number of potentially relevant input variables by using suitable priors with hyperparameters controlling the effect of the inputs in the model (see, e.g., Lampinen & Vehtari, 2001). Although such models may have good predictive performance, it may be difficult to analyse them, or costly to make measurements or computations. To make the model more explainable (easier to gain scientific insights) or to reduce the measurement cost or the computation time, it may be useful to select a smaller set of input variables. In addition, if the assumptions of the model and prior do not match well the properties of the data, reducing the number of input variables may even improve the performance of the model.

In prediction and decision problems, it is natural to assess the predictive ability of the model by estimating the expected utilities, as the principle of rational decisions is based on maximizing the expected utility (Good, 1952; Bernardo & Smith, 1994) and the maximization of expected likelihood maximizes the information gained (Bernardo, 1979). In machine learning community expected utility is sometimes called generalization error. Following simplicity postulate (Jeffreys, 1961), it is useful to start from simpler models and then test if more complex model would give significantly better predictions. Combining the principle of rational decisions and simplicity postulate, our goal is to find a model with the smallest number of input variables having statistically or practically at least the same predictive ability as the full model with all the available inputs. An additional advantage of comparing the expected utilities is that it takes into account the knowledge of how the model predictions are going to be used and further it may reveal that even the best model selected from some collection of models may be inadequate or not practically better than the previously used models.

Vehtari and Lampinen (2002, 2003) present with Bayesian justification how to obtain distributions of expected utility estimates of complex Bayesian hierarchical models using cross-validation predictive densities. The distribution of the expected utility estimate describes the uncertainty in the estimate and 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. In the case of K inputs, there are 2^K input combinations, and computing the expected utilities for each model easily becomes computationally prohibitive. To use expected utility approach we need to find way to find out smaller number of potentially useful input combinations. One approach would be using global optimization algorithms to search the input combination maximizing the expected utility (e.g. Draper & Fouskakis, 2000). Potential problems with this approach are that it may be slow, requiring hundreds or thousands of cross-validation evaluations, and results depend on search heuristics

Current trend in Bayesian model selection (including the input variable selection) is to estimate posterior probabilities of the models using Markov chain Monte Carlo (MCMC) methods and especially variable dimension MCMC methods (Green, 1995; Stephens, 2000). The variable dimension MCMC visits models according to their posterior probabilities, and thus models with negligible probability are probably not visited in finite time. The posterior probabilities of the models are then estimated based on number of visits for each model and models with highest posterior probabilities are further investigated.

Marginal likelihoods and posterior probabilities of the input combinations have been used directly for input selection, for example, by Brown, Vannucci, and Fearn (1998), Ntzoufras (1999), Han and Carlin (2000), Sykacek (2000), Kohn, Smith, and Chan (2001), and Chipman, George, and McCulloch (2001). Although this kind of approach has produced good results, it may be sensitive to prior choices as discussed in section 2.4, and it does not necessarily provide model with the best expected utility as demonstrated in section 3. Furthermore, Spiegelhalter (1995) and Bernardo and Smith (1994) argue that

when selecting a single model from some family of models instead of integrating over the discrete model choices (e.g., input combinations), it is better to compare the consequences (e.g., utilities) of the models instead of their posterior probabilities.

We propose to use posterior and marginal posterior probabilities obtained via variable dimension MCMC methods to find out potentially useful input combinations and to do the final model choice and assessment using the expected utilities (with any desired utility) computed by using the cross-validation predictive densities. Similar approach has been used before at least by Vannucci, Brown, and Fearn (2001), and Brown, Vannucci, and Fearn (2002), but they considered only general linear models and squared loss, and they did not estimate the uncertainty in the expected loss estimates.

As illustrative examples, we use MLP networks (MLP) and Gaussian processes (GP) (Neal, 1996, 1999; Lampinen & Vehtari, 2001) in one toy problem and two real world problems: concrete quality estimation and forest scene classification. MLPs and GPs are non-parametric non-linear models, where the interactions between the input variables are handled implicitly in the model and thus there is no need to specify them explicitly. This alleviates the input variable selection, as we only need to select whether the input variable should be included in the model or not, and if it is included, the model automatically handles possible interactions. For MLPs and GPs it is common to use hierarchical prior structures which produce continuous input variable selection, that is, there are hyperparameters which control how strong effect each input variable may have in the model. Consequently, model averaging (averaging over different input variable combinations) usually does not produce different results than the the model with all the input variables (full model), and thus we may use the full model as the baseline, to which we can compare models with fewer input variables.

We have tried to follow the notation of Gelman, Carlin, Stern, and Rubin (1995) and we assume that reader has basic knowledge of Bayesian methods (see, e.g., short introduction in Lampinen & Vehtari, 2001) and Bayesian model assessment and comparison based on expected utilities (Vehtari & Lampinen, 2002, 2003). Knowledge of MCMC, MLP or GP methods is helpful but not necessary.

2 Methods

We first discuss relationship of posterior probabilities of models to expected utilities (section 2.1). Then we discuss variable dimension MCMC methods, which can be used to obtain posterior probability estimates for a large number of models in a time comparable to computing the cross-validation predictive densities for a single model (section 2.2). Finally we discuss prior issues specific in input selection (sections 2.3 and 2.4).

2.1 Expected utilities and posterior probabilities

Given models M_l, $l = 1, \ldots, L$ we would like to find the model with the smallest number of input variables having at least the same expected utility as the full model with all the available inputs. One way would be to estimate expected utility $\bar{u}_{M_l}$ for each model, but this may be computationally prohibitive.

Distributions of expected utility estimates of complex Bayesian hierarchical models can be obtained using cross-validation predictive densities (Vehtari & Lampinen, 2002, 2003). Using the distributions it is easy to compute the the probability of one model having a better expected utility than some other model $p(\bar{u}_{M_1-M_2} > 0)$. In real world problems it is useful to use application-specific utilities for model assessment. For example, Draper and Fouskakis (2000) discuss example in which monetary utility is used for data collection costs and the accuracy of predicting mortality rate in health policy problem. However, for model comparison likelihood based utilities are useful as they measure how well the model

estimates the whole predictive distribution. Expected predictive likelihoods are also related to posterior probabilities.

The posterior predictive distribution of not yet observed $y^{(n+1)}$ given the training data $D = \{y^{(i)}; i = 1, 2, \ldots, n\}$, where $y^{(i)}$ are i.i.d., is obtained by integrating the predictions of the model with respect to the posterior distribution of the model

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

where θ denotes all the model parameters and hyperparameters and I denotes the assumptions about the model space discussed in the section 2.3. For convenience we consider expected predictive log-likelihood which is given by

$$E_{y^{(n+1)}} \left[\log p(y^{(n+1)}|D, M_l, I)\right]. \tag{2}$$

Assuming that the distribution of the $y^{(n+1)}$ is same as the distribution of $y^{(i)}$, Equation 2 can be approximated using the cross-validation predictive densities as

$$E_i \left[\log p(y^{(i)}|D^{(\setminus i)}, M_l, I)\right], \tag{3}$$

where $D^{(\setminus i)}$ denotes all the elements of D except $y^{(i)}$. Samples from cross-validation predictive densities are easily obtained with importance-sampling leave-one-out CV or k-fold-CV.

Commonly used Bayesian model selection method is the Bayes factor (Kass & Raftery, 1995)

$$B_{jk} = \frac{p(D|M_j, I)}{p(D|M_k, I)}, \tag{4}$$

where

$$p(D|M_l, I) = \prod_i p(y^{(i)}|y^{(s_i)}, M_l, I), \tag{5}$$

where the right hand side is obtained using the chain rule, and $y^{(s_i)}$ is a set of data points so that $y^{(s_1)} = \emptyset$, $y^{(s_2)} = y^{(1)}$, and $y^{(s_i)} = y^{(1,\ldots,i-1)}$; $i = 3, \ldots, n$. $p(D|M_l, I)$ is called marginal probability of the data, or prior predictive likelihood. Taking the logarithm of the right hand side of the Equation (5) and dividing by n we get

$$E_i \left[\log p(y^{(i)}|y^{(s_i)}, M)\right], \tag{6}$$

which is similar to cross-validation estimate of the expected predictive likelihood (Equation 3). In expected utility sense this is an average of predictive log-likelihoods with number of data points used for fitting ranging from 0 to $n - 1$. As the learning curve is usually steeper with smaller number of data points, this is less than the expected predictive log-likelihood with $n/2$ data points. As there are terms which are conditioned on none or very few data points, Equation 6 is sensitive to prior changes. With more vague priors and more flexible models few first terms dominate the expectation unless n is very large.

This comparison shows that the prior predictive likelihood of the model can be used as lower bound of the expected predictive likelihood (favoring less flexible models). The predictive likelihood is good utility for model comparison, but naturally the final model choice and assessment can be done using the application specific utilities. The prior predictive likelihoods can also be combined with prior probabilities of the models getting posterior probabilities of the models (with uniform prior on models prior predictive likelihoods are proportional to posterior probabilities). It is possible to use the prior predictive likelihoods or the posterior probabilities depending whether it is believed that using non-uniform prior on model space helps us to find more useful models. Model space prior affects only which models are

selected for further study, it does not affect the final model comparison based on expected utilities. In our examples we have used uniform prior or modest prior favoring smaller models (see discussion on section 2.3 and examples in section 3).

Computation of the prior predictive likelihood $p(D|M_l, I)$ for complex hierarchical models is usually very hard (Kass & Raftery, 1995). In the next section we discuss variable dimension MCMC methods which can be used to estimate posterior probabilities $p(M_l, I|D)$ and as

$$p(M_l, I|D) = p(D|M_l, I)p(M_l|I)/p(D|I) \tag{7}$$

it is also possible to obtain relative prior predictive likelihood values ignoring $p(D|I)$ which is constant for all models M_l given I.

If there are many correlated inputs, it is probable that there are also many high-probability input combinations and thus it may be hard to estimate the probabilities of input combinations well. In this case, we propose to use the marginal probabilities of the inputs, which are easier to estimate, to indicate potentially useful inputs. This is illustrated in sections 3.4 and 3.6.

In addition to input selection, the marginal probabilities of inputs can be used to estimate the relevance of the inputs, which has great importance in analyzing the final model. For MLP networks, MacKay (1994), and Neal (1996) proposed the "automatic relevance determination" (ARD) prior as an automatic method for determining the relevance of the inputs in MLP. In section 3.4 we discuss and illustrate the benefit of the marginal probabilities of inputs over the ARD values for relevance estimation in MLP.

2.2 Variable dimension Markov chain Monte Carlo

For simple models it may be possible to estimate the prior predictive likelihood $p(D|M_l, I)$ analytically, but even in this case if there is large number of models it may be computationally prohibitive to do it for all models. In case of complex hierarchical models we usually do not have analytic solutions or good analytical approximations (e.g., variational approximations, Jordan, Ghahramani, Jaakkola, & Saul, 1998), and we need to resort to stochastic approximations such as the the Monte Carlo methods (Gilks, Richardson, & Spiegelhalter, 1996; Robert & Casella, 1999; Chen, Shao, & Ibrahim, 2000). In the MCMC, samples from the posterior distribution are generated using a Markov chain that has the desired posterior distribution as its stationary distribution.

In the case of input selection, models will generally have different numbers of parameters if they have different number of inputs. The variable dimension MCMC methods (Green, 1995; Stephens, 2000; Ntzoufras, 1999; Han & Carlin, 2000) allow jumps between models with different dimensional parameter spaces, and thus we can get samples from the posterior distribution of the input combinations. The variable dimension MCMC methods visits models (one visit is one sample) according to their posterior probabilities, and thus models with negligible probability are probably not visited in finite time. Consequently, only the most probable models are investigated and computational savings can be considerable compared to going through all possible models. Speed and accuracy of variable dimension MCMC methods may in some cases further increased by analytically marginalizing over as many parameters as possible.

We have used the reversible jump Markov chain Monte Carlo (RJMCMC; Green, 1995) which is is one of the simplest to implement and one of the fastest on big problems. The RJMCMC is an extension to the Metropolis-Hastings method allowing jumps between models with different dimensional parameter spaces. In the case of input selection, models have different number of parameters as they have different number of inputs. When adding or removing inputs, the corresponding parameters are added or removed,

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

$$\alpha = \min\left(1, \frac{p(D|\theta_{M_2}, M_2)\, p(\theta_{M_2}|M_2)\, p(M_2|I)\, j(M_2, M_1) q(u_2|\theta_{M_2}, M_2, M_1)}{p(D|\theta_{M_1}, M_1)\, p(\theta_{M_1}|M_1)\, p(M_1|I)\, j(M_1, M_2) q(u_1|\theta_{M_1}, M_1, M_2)} \left|\frac{\partial h_{M_1,M_2}(\theta_{M_1}, u_1)}{\partial(\theta_{M_1}, u_1)}\right|\right), \tag{8}$$

where I denotes the assumptions about the model space, j is the probability of jumping from one model to another, q is the proposal distribution for u and h_{M_1,M_2} is an invertible function defining mapping $(\theta_{M_2}, u_2) = h_{M_1,M_2}(\theta_{M_1}, u_1)$.

In the case of suitable proposal distribution, the acceptance probability term can be greatly simplified. When adding a new input, we set h_{M_1,M_2} as identity, that is $(\theta_{M_2}) = (\theta_{M_1}, u_1)$, and then use the conditional prior of the new parameters as the proposal distribution (see sections 3.1 and 3.2). Now the Jacobian is 1, the prior terms for the parameters common to both models cancel out and the prior and the proposal distribution for the new parameters cancel out. Moreover, as we set $j(M_1, M_2) = j(M_2, M_1)$, Equation 8 simplifies to

$$\alpha = \min\left(1, \frac{p(D|\theta_{M_2}, M_2)\, p(M_2|I)}{p(D|\theta_{M_1}, M_1)\, p(M_1|I)}\right). \tag{9}$$

We use hierarchical priors for the parameters specific to inputs, and so the conditional prior of the new parameters is natural proposal distribution with a reasonable acceptance rate and mixing behaviour. In our case studies time needed for obtaining posterior probability estimates for all input combinations (or marginal probability estimates for inputs) with RJMCMC was relative to time used for computing expected utilities with k-fold-CV predictive densities for single model.

2.3 Priors on model space

Model space prior can be used to favor certain models or even if we would like to use uniform prior on models, it may be useful to use non-uniform prior due to computational reasons. As discussed below it is possible that when using uniform prior on models the implicit prior on number of inputs may cause the variable dimension MCMC methods to miss some input combinations. Using non-uniform prior we may be able to improve sampling and to obtain posterior probabilities based on uniform prior, the appropriate prior correction can be made afterwards.

We are interested in input variable selection for MLPs and GPs, where the interactions between the input variables are handled automatically in the model, and thus we do not need to consider model space priors which consider the interactions. For example, Chipman et al. (2001) discuss some choices for model space priors in the case of explicit interactions in the model.

If we have K input variables, there are $L = 2^K$ possible different input combinations (models). A simple and popular choice is the uniform prior on models

$$p(M_l) \equiv 1/L\,, \tag{10}$$

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

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

$$\gamma = (\gamma_1, \ldots, \gamma_K)^T, \tag{11}$$

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

$$p(\gamma) = (1/2)^K. \tag{12}$$

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

$$p(k) = \text{Bin}(K, 1/2), \tag{13}$$

which clearly is not noninformative, as $E[k] = 0.5K$ and $\text{Var}[k] = 0.25K$. For example, if K=27, then k lies in the range 7 to 20 with prior probability close to 1, and thus it is possible that variable dimension MCMC methods will not sample models with less than 7 inputs (see also examples in section 3).

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

$$p(k) = \text{Bin}(K, 1/2)h(k), \tag{14}$$

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

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

$$p(\gamma) = \pi^k(1-\pi)^{1-k} \tag{15}$$

and correspondingly

$$p(k) = \text{Bin}(K, \pi). \tag{16}$$

In this case, $E(k|\pi) = K\pi$ and $\text{var}(k|\pi) = K\pi(1-\pi)$. Although having more control, this would still effectively restricts k *a priori* to lie in a short range

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

$$p(\pi) = \text{Beta}(\alpha, \beta), \tag{17}$$

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

$$p(k) = \text{Beta-bin}(n, \alpha, \beta). \tag{18}$$

In this case, $E[k|\pi, \alpha, \beta] = K\frac{\alpha}{\alpha+\beta}$ and $\text{Var}[k|\pi, \alpha, \beta] = K\frac{\alpha\beta(\alpha+\beta+K)}{(\alpha+\beta)^2(\alpha+\beta+1)}$, and thus the values for α and β are easy to solve after setting $E[k]$ and $\text{Var}[k]$ to the desired values. As the Beta-binomial is often nonsymmetric, it may be easier to choose the values for α and β by plotting the distribution with different values of α and β, as we did in the examples in section 3. If $\alpha = 1$ and $\beta = 1$ then the prior on k is uniform distribution on $(0, K)$, but now the models are not equally probable, as the models with few or many inputs have higher probability than the models with about $K/2$ inputs. For example, for models with more than $K/2$ inputs, the model with one extra input is a priori K times more probable. Consequently, it is not possible to be uninformative in input selection, and some care should be taken when choosing priors, as efforts to be uninformative in one respect will force one to be informative in other respect. Even if there is some prior belief about the number of inputs, it may be hard to present in mathematical form or there may be computational problems as in the example in section 3.6.

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

$$p(\gamma) = \prod \pi_k^{\gamma_k}(1-\pi_k)^{1-\gamma_k}, \tag{19}$$

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

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

2.4 Priors on input specific parameters

As discussed and illustrated, for example, by Richardson and Green (1997), Dellaportas and Forster (1999), and Ntzoufras (1999), the prior on parameters $p(\theta_{M_l}|M_l)$ greatly affects the prior predictive likelihood (and thus posterior probability) of the model M_l having extra parameter $\theta^+_{M_l}$. If the prior on the extra parameters $p(\theta^+_{M_l}|M_l)$ is too tight, the extra parameters might not reach a useful range in the posterior, thus making the model less probable. On the other hand, if the prior is too vague, the probability of any value for the extra parameter gets low, and correspondingly, the probability of the model gets low.

Often, it is recommended to test different priors, but there is no formal guidance what to do if the different priors produce different results. Some methods for controlling the effects of the prior in linear models are discussed by Ntzoufras (1999), but these methods may be difficult to generalize to other models. Using hierarchical priors seems to alleviate partly the problem, as discussed by Richardson and Green (1997) and illustrated in section 3. Furthermore, since the effect of the model space prior is considerable and its selection usually quite arbitrary, there is probably no need to excessively fine tune the priors of the parameters in question. Naturally, the prior sensitivity is an even lesser problem when the final model choice is based on the expected utilities.

3 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; Vehtari, 2001) in one toy problem (section 3.4) and two real world problems: concrete quality estimation (section 3.5) and forest scene classification (section 3.6). We first briefly describe the models and algorithms used (sections 3.1, 3.2, and 3.3).

3.1 MLP neural networks

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

$$f(x, \theta_w) = b_2 + w_2 \tanh(b_1 + w_1 x),$$

where θ_w denotes all the parameters w_1, b_1, w_2, b_2, which are the hidden layer weights and biases, and the output layer weights and biases, respectively. We used Gaussian priors on weights and the Automatic

Relevance Determination (ARD) prior on input weights

$$
\begin{aligned}
w_{1,kj} &\sim N(0, \alpha_{w_1,k}),\\
\alpha_{w_1,k} &\sim \text{Inv-gamma}(\alpha_{w_1,\text{ave}}, \nu_{w_1,\alpha})\\
\alpha_{w_1,\text{ave}} &\sim \text{Inv-gamma}(\alpha_{w_1,0}, \nu_{w_1,\alpha,\text{ave}})\\
\nu_{w_1,\alpha} &= V[i]\\
i &\sim U_d(1, K)\\
V[1:K] &= [0.4,\ 0.45,\ 0.5:0.1:1.0,\ 1.2:0.2:2,\ 2.3,\ 2.6,\ 3,\ 3.5,\ 4]\\
b_1 &\sim N(0, \alpha_{b_1})\\
\alpha_{b_1} &\sim \text{Inv-gamma}(\alpha_{b_1,0}, \nu_{\alpha,b_1})\\
w_2 &\sim N(0, \alpha_{w_2})\\
\alpha_{w_2} &\sim \text{Inv-gamma}(\alpha_{w_2,0}, \nu_{\alpha,w_2})\\
b_2 &\sim N(0, \alpha_{b_2})
\end{aligned}
$$

where the α's are the variance hyperparameters ($\alpha_{w_1,k}$'s are also called ARD parameters), ν's are the number of degrees of freedom in the inverse-Gamma distribution. To allow easier sampling with Gibbs method, discretized values of ν were used so that $[a : s : b]$ denotes the set of values from a to b with step s, and $U_d(a, b)$ is a uniform distribution of integer values between a and b.

Since the ARD prior allows less relevant inputs to have smaller effect in the model, it produces effect similar to continuous input variable selection, and thus discrete input variable selection is not always necessary. To be exact, the ARD prior controls the nonlinearity of the input, instead of the predictive importance or causal importance (Lampinen & Vehtari, 2001), but since "no effect" is linear effect, it will also allow irrelevant inputs to have smaller effect in the model. This is illustrated in section .

In MLP, the weights $w_{1,kj}$ are connected to input k. For a new input k, the $w_{1,kj}$ and $\alpha_{w_1,k}$ were generated from the proposal distribution (see Equation 8), which was the same as their respective conditional prior distributions. As a hierarchical prior structure was used, the conditional priors were also adapting to the data and so useful acceptance rates were obtained. We also tested the conditional maximization and the auxiliary variable methods (Brooks, Giudici, & Roberts, 2003). Finding the conditional maximum was too slow and unstable while the auxiliary variable method easily got stuck in, despite of tuning attempts.

3.2 Gaussian processes

The Gaussian process 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

$$
\begin{aligned}
E_y[y|x^{(n+1)}, \theta, D] &= k^{\mathrm{T}} C^{-1} y^{(1,\ldots,n)}\\
\text{Var}_y[y|x^{(n+1)}, \theta, D] &= V - k^{\mathrm{T}} C^{-1} k,
\end{aligned}
$$

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

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

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

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

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

Similar to MLP, in GP the "ARD" parameters ρ_u measure the nonlinearity of the inputs as ρ_u defines the characteristic length of the function for given input direction. However, this prior produces effect similar to continuous input variable selection, and thus discrete the input variable selection is not always necessary.

For a new input, the corresponding ρ_u was generated from its conditional prior. As a hierarchical prior structure was used, the conditional prior was also adapting and so useful acceptance rates were obtained. The acceptance rates for the GP were naturally higher than for MLP as proposal distribution was univariate compared to $J + 1$-dimensional proposal distribution for MLP (J is number of hidden units). We also tested the auxiliary variable method (Brooks et al., 2003), but it did not improve acceptance rates, despite of some tuning attempts.

3.3 Algorithms used

For integration over model parameters and hyperparameters the in-model sampling was made using Metropolis-Hastings sampling, Gibbs sampling, and hybrid Monte Carlo (HMC) as described in References (Neal, 1996, 1997, 1999; Vehtari & Lampinen, 2001; Vehtari, 2001) and for estimation of posterior probabilities of input combinations the between-model sampling was made using the RJMCMC as described in References (Green, 1995; Vehtari, 2001). Between-model jump consisted of proposing adding or removing single input or switching included input to not-included input, and the proposal distribution for the new parameters was the conditional prior of the new parameters. As hierarchical priors for the parameters specific to inputs were used, the conditional priors are adapting to the data and thus the conditional prior is a natural proposal distribution with a reasonable acceptance rate and mixing behavior. The MCMC sampling was done with the FBM[1] software and Matlab-code partly derived from the FBM and Netlab[2] toolbox.

To make convergence diagnostics and estimation of credible intervals (CI) easier, ten independent RJMCMC chains (with different starting points) were run for each case. For convergence diagnostics, we used visual inspection of trends, the potential scale reduction method (Gelman, 1996) and the Kolmogorov-Smirnov test (Robert & Casella, 1999). For between-model convergence diagnostics, we used the chi-squared and Kolmogorov-Smirnov tests proposed by Brooks, Giudici, and Philippe (2002), which also utilize several independent chains. As the number of visits to each model was typically very

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

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

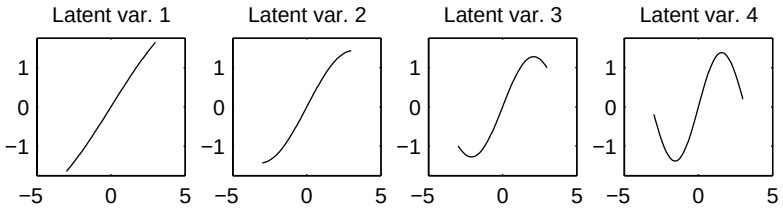

Figure 1: The target function is an additive function of four inputs. The predictive importance of every input is equal, in RMSE terms, as the latent functions are scaled to equal variance over the uniform input distribution $U(-3, 3)$.

low, we mainly analysed the visits to each subpopulation having equal number of inputs. Other convergence assessment methods for the RJMCMC are discussed, for example, by Brooks and Giudici (1999), and Brooks and Giudici (2000). Sequential correlations of MCMC samples were estimated using autocorrelations (Neal, 1993; Chen et al., 2000) and chains were thinned to get less dependent samples. Depending on case every 400th–1600th meta-iteration sample was saved and total of 4000 samples were saved from ten independent chains.

The distributions of the expected utilities of the models were estimated using the cross-validation predictive densities obtained using k-fold-CV as described by Vehtari and Lampinen (2002).

3.4 Toy problem

With suitable priors it is possible to have a large number of input variables in Bayesian models, as less relevant inputs can have a smaller effect in the model. For example, in MLP it is useful to use so called "automatic relevance determination" prior (ARD; MacKay, 1994; Neal, 1996). In ARD each group of weights connected to the same input has common variance hyperparameters, while the weight groups can have different hyperparameters. In many experiments ARD has been shown to be useful to allow many input variables in the MLP (Lampinen & Vehtari, 2001). Such models may have good predictive performance, but it may be difficult to analyse them, or costly to make measurements or computations, and thus input selection may be desirable.

In this toy problem we compare ARD, posterior probabilities of inputs and expected utility based approach for input relevance determination. The target function is an additive function of four inputs (Figure 1), with equal predictive importance for every input. We used 10-hidden-unit MLPs with similar priors as described in (Lampinen & Vehtari, 2001; Vehtari, 2001) and uniform prior on all input combinations

Figure 2 shows the predictive importance, posterior probability, the mean absolute values of the first and second order derivatives of the output with respect to each input, and the relevance estimates from the ARD. Note, that the ARD coefficients are closer to the second derivatives than to the first derivatives (local causal importance) or to the error due to leaving input out (predictive importance).

Figure 3 shows the predictive importance, the relevance estimates from the ARD, and posterior probabilities of inputs 1 (almost linear) and 4 (very nonlinear) when the weighting of the input 4 is varied from 1 to 1/16. As the inputs are independent, the leave-input-out error measures well the predictive importance. Based on the ARD values, it is not possible to know which one of the inputs is more important. It is also hard to distinguish the irrelevant input from almost linear input. Marginal probability of the input indicates well whether the input has any predictive importance.

Figure 4 shows the predictive importance, posterior probabilities, and the relevance estimates from

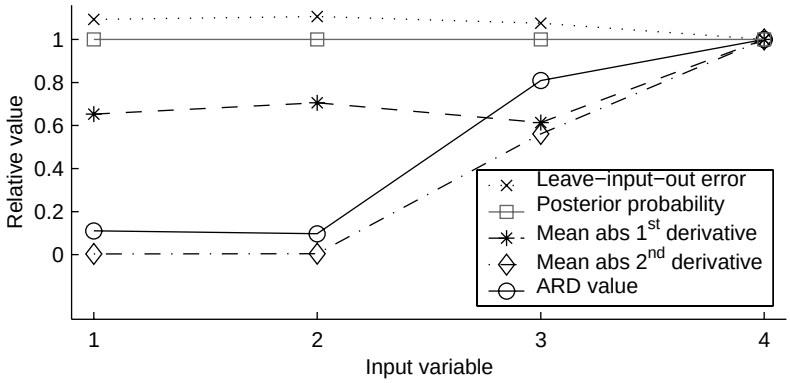

Figure 2: Different measures of importance of inputs for the test function in Figure 1. The ARD coefficients are closer to the second derivatives than to the first derivatives (local causal importance) or to the error due to leaving input out (predictive importance).

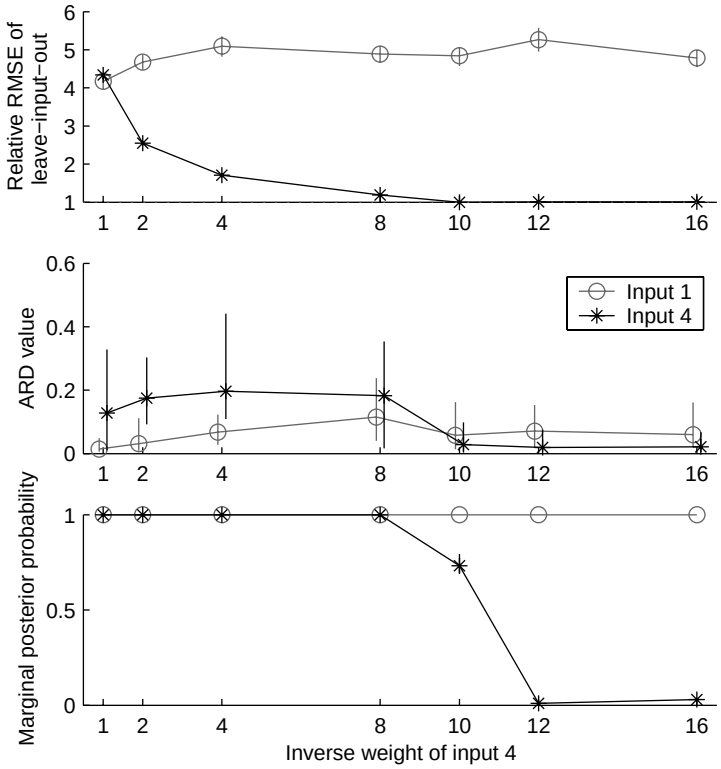

Figure 3: Top plot shows the leave-input-out error. Middle plot shows that although weight of the input 4 is reduced, ARD value stays at constant level measuring the nonlinearity of the input, until the weight is so small that the information from the input is swamped by the noise. Bottom plot plot shows that the probability of the input indicates whether the input has any predictive importance.

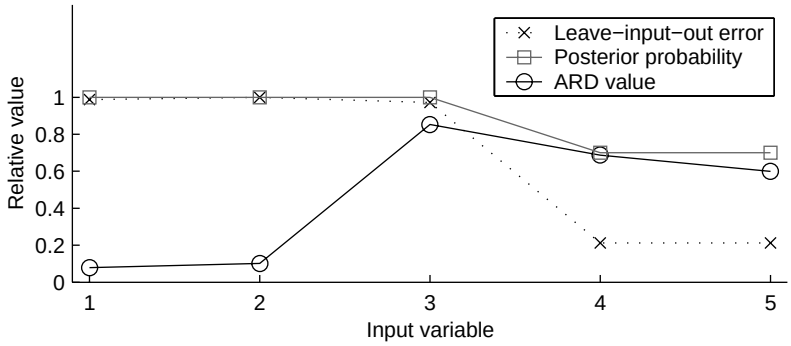

Figure 4: Input 5 is duplicate of input 4. Low leave-input-out error for inputs 4 and 5 does not mean that both inputs could be left out. Posterior probabilities and ARD values for both inputs 4 and 5 are smaller than in the model without the input 5 (see Figure 2), but the correlation between inputs can be diagnosed (see text).

Input 5	Input 4 = 0	Input 4 = 1	
0	0.0	0.3	0.3
1	0.3	0.4	0.7
	0.3	0.7	

(a) Estimated with RJMCMC

Input 5	Input 4 = 0	Input 4 = 1	
0	0.1	0.2	0.3
1	0.2	0.5	0.7
	0.3	0.7	

(b) Assuming independence

Input 5	Input 4 = 0	Input 4 = 1
0	0.0	1.4
1	1.4	0.8

(c) Relative difference

Table 1: Joint and marginal posterior probabilities of inputs 4 and 5. 0 and 1 indicate whether the input is in the model. Note that at least one of the inputs has to be in the model but preferably only one of them.

the ARD in case where the input 5 is duplicate of input 4. Leave-input-out error indicates that either of the inputs 4 or 5 could be left out, but in order to know if both could be left out backward elimination should be used. Marginal posterior probabilities and ARD values for single inputs are lower than in the case without the 5th input, but now it is possible to examine joint probabilities of inputs and correlations of ARD values. ARD values of input 4 and 5 have correlation coefficient of -0.68, which indicates that maybe only one of the inputs is necessary. Tables 1a,b,c illustrate the effect of correlation to the posterior probabilities and indicate well that it is necessary to have at least one the inputs in the model and preferably only one of them.

When the inputs are independent, the leave-input-out error measures well the predictive importance. The ARD does not measure the predictive importance and thus it does not distinguish an irrelevant input from an almost linear input. Marginal probability of the input indicates well whether the input has any predictive importance. When the inputs are dependent, the leave-input-out does not measure the dependence between inputs. To handle dependencies it could be replaced with computationally heavy backward elimination. Although the ARD does not measure the predictive importance, the correlations between inputs can be discovered by examining the correlations between the ARD values. Joint predictive importances and dependencies of inputs can be easily diagnosed by examining the joint probabilities of inputs.

3.5 Real world problem I: Concrete quality estimation

In this section, we present results from the 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), flow value, 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 selected by the concrete expert, (listed, e.g., in Figure 7) and 215 samples designed to cover the practical range of the variables, collected by the concrete manufacturing company. In the following, we report results for the *air-%*.Similar results were achieved for other target variables (Vehtari, 2001). See the details of the problem, the descriptions of the variables and the conclusions made by the concrete expert in Reference (Järvenpää, 2001).

The aim of the study was to identify which properties of the stone material are important, and additionally, examine the effects that properties of the stone material have on concrete. It was desirable to get both the estimate of relevance of all available input variables and select a minimal set required to get a model with statistically the same predictive capability as with the full model. A smaller model is easier to analyze and there is no need to make possibly costly or toxic measurements in the future for properties having negligible effect. Note that as the cost of the toxic measurements was difficult to estimate, we did not include the cost directly to the utility function. Instead we just tested which toxic measurements could be left out without statistically significant drop in prediction accuracy. The problem is complicated because there are strong cross-effects, and the inputs measuring similar properties have strong dependencies.

For models used in (Järvenpää, 2001), we had made the input selection using the deviance information criterion (DIC) (Spiegelhalter, Best, Carlin, & van der Linde, 2002) and heuristic backward selection. DIC estimates the expected utility using plug-in predictive distribution and asymptotic approximation (Vehtari, 2002; Vehtari & Lampinen, 2003). Although this approach produced reasonable results, it required a full model fitting for each model investigated, contained some *ad hoc* choices to speed up the heuristic backward selection, and lacked estimate of the associated uncertainty and clear results for the relevance of the different inputs.

Below we present results using the RJMCMC and the expected utilities computed by using the cross-validation predictive densities. With this approach, we were able to get more insight about the problem, smaller models, and improved reliability of the results. We used Gaussian process models with quadratic covariance function and ARD type hierarchical prior for input specific parameters. The residual model used was input dependent Student's t_ν with unknown degrees of freedom ν. As the size of the residual variance varied depending on three inputs, which were zero/one variables indicating the use of additives, the parameters of the Student's t_ν were made dependent on these three inputs with a common hyperprior.

From about 10^8 possible input combinations, the 4000 saved states included about 3500 and 2500 different input combinations with uniform and Beta priors, respectively. Few most probable models were visited by all ten independent chains and for example, ten most probable models were visited by at least eight chains. Thus, useful credible intervals could be computed for the model probabilities.

Figures 5 and 6 show the posterior probabilities of the number of inputs with an equal prior probability for all the models and with Beta-bin(27, 5, 10) prior on the number of inputs, respectively. With equal prior probability for all models, the prior probability for the number of inputs being less than eight is so low that it is unlikely that the RJMCMC will visit such models. Parameters for the Beta-binomial prior were selected to better reflect our prior information, that is, we thought it might be possible to have

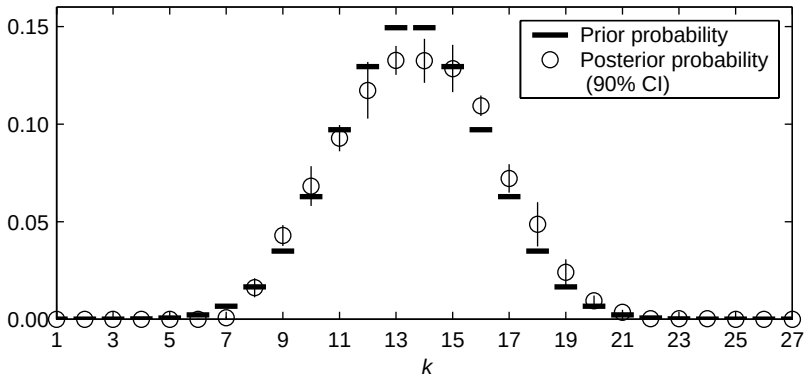

Figure 5: Concrete quality estimation example, predicting the air-% with GP: The posterior probabilities of the number of inputs with "uninformative" prior, i.e., equal prior probability for all models.

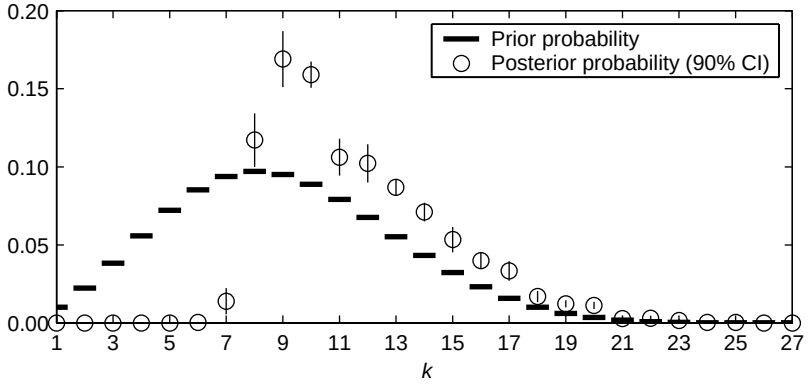

Figure 6: Concrete quality estimation example, predicting the air-% with GP: The posterior probabilities of the number of inputs with Beta-bin(27, 5, 10) prior on the number of inputs.

a low number of inputs, most probably about 6-12 inputs and not excluding the possibility for a larger number of inputs. Note that the Beta-binomial prior used is in fact more vague about the number of inputs than the "uninformative" prior. The posterior distribution of the number of inputs is quite widespread, which is natural as the inputs are dependent and the ARD type prior allows use of many inputs.

Figure 7 shows the marginal posterior probabilities of the inputs with a Beta-bin(27,5,10) prior on the number of inputs. The nine most probable inputs are clearly more probable than the others and the other inputs have posterior probability approximately equal to or less than the mean prior probability of an input (1/3).

Figure 8 shows the ARD values of the inputs for the full model. Eight of the nine most probable inputs have also a larger ARD value than the other inputs, but they cannot be clearly distinguished from the other inputs. Moreover, input *"BET"* (measuring the specific surface area of the fines) is ranked much lower by the ARD than by the probability (compare to Figure 7). Further investigation revealed that *"BET"* was relevant, but had near linear effect. Figure 9 shows the posterior probabilities of the ten most probable input combinations with a Beta-bin(27,5,10) prior on the number of inputs. All the ten models are very similar, only minor changes are present in few inputs, and, the changed inputs are known to correlate strongly. In this case, two models are significantly more probable than others, but between

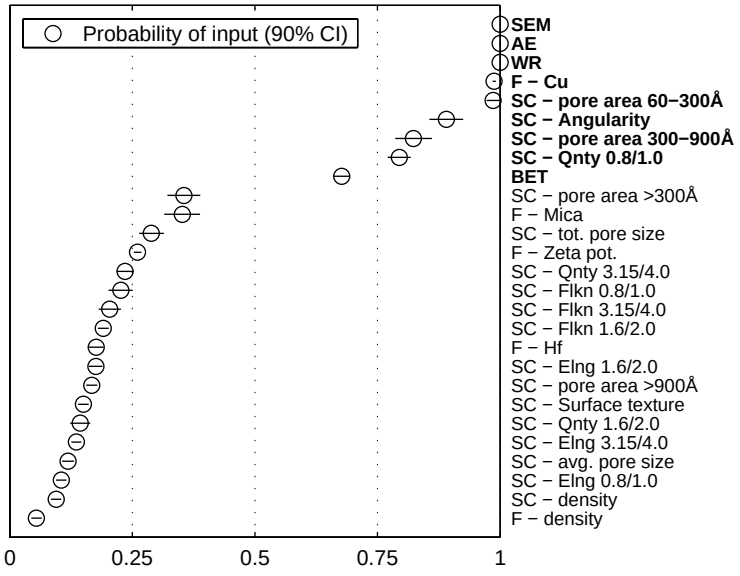

Figure 7: Concrete quality estimation example, predicting the air-% with GP: The marginal posterior probabilities of the inputs with a Beta-bin(27,5,10) prior on the number of inputs. The inputs in the most probable model are in boldface (see Figure 9).

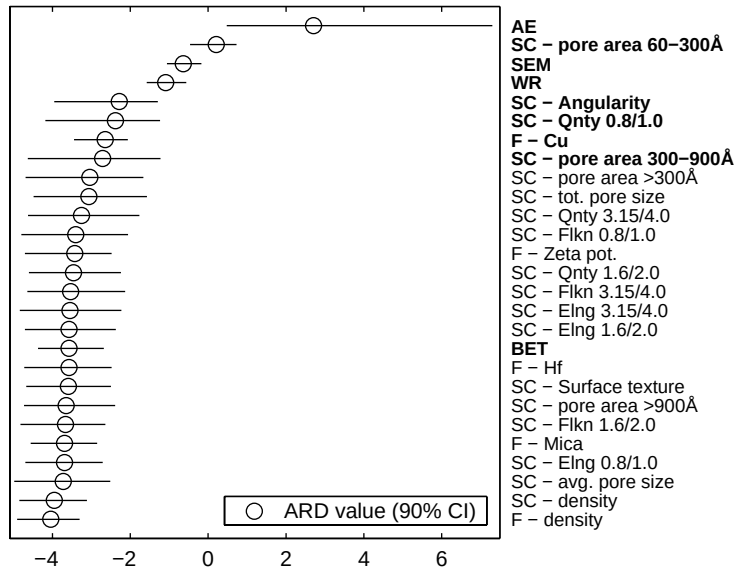

Figure 8: Concrete quality estimation example, predicting the air-% with GP: The ARD values of the inputs of the full model. The nine most probable inputs are in boldface. Compare to Figure 7.

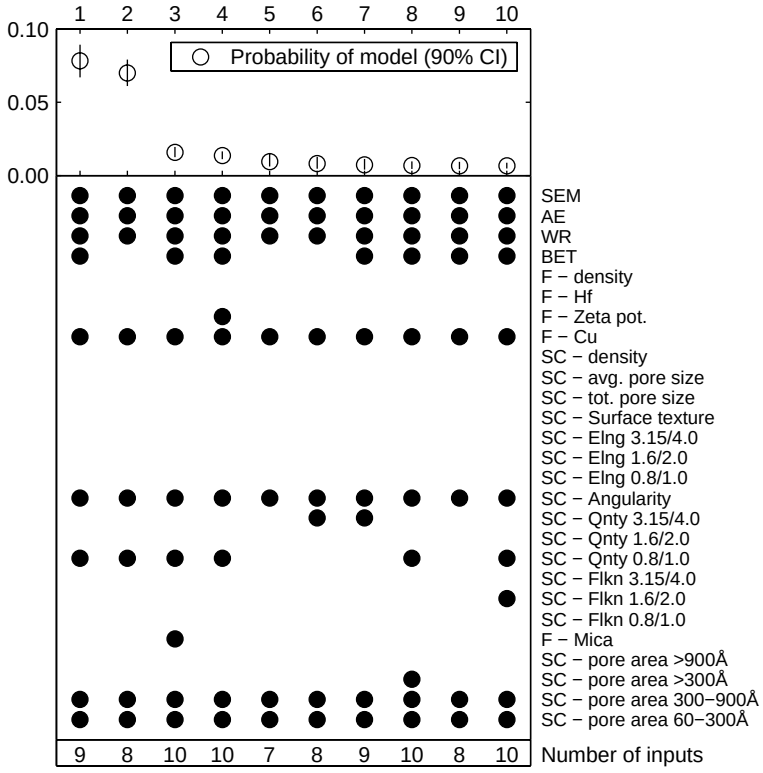

Figure 9: Concrete quality estimation example, predicting the air-% with GP: The probabilities of the ten most probable models with a Beta-bin(27,5,10) prior on the number of inputs. The top part shows the probabilities of the models, the middle part shows which inputs are in the model, and the bottom part shows the number of inputs in the model.

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

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

To illustrate the differences between the posterior probabilities and the expected predictive likelihoods, Figure 10 shows the expected utilities computed using the cross-validation predictive densities for the full model and the models having the k ($k = 5, \ldots, 15$) most probable inputs. Note that the expected predictive likelihoods are similar for models having at least about eight most probable inputs, while posterior probabilities are very different for models with different number of inputs. For example, the posterior probability of the full model is vanishingly small compared to the most probable models, but the expected predictive likelihood is similar to the most probable models. The performance of the full model is similar to smaller models, as the ARD type prior allows many inputs without reduced predictive performance. Note that if the point estimate (e.g., mean) of the expected utility would be used for model selection, larger models would be selected than when selecting the smallest model with statistically the

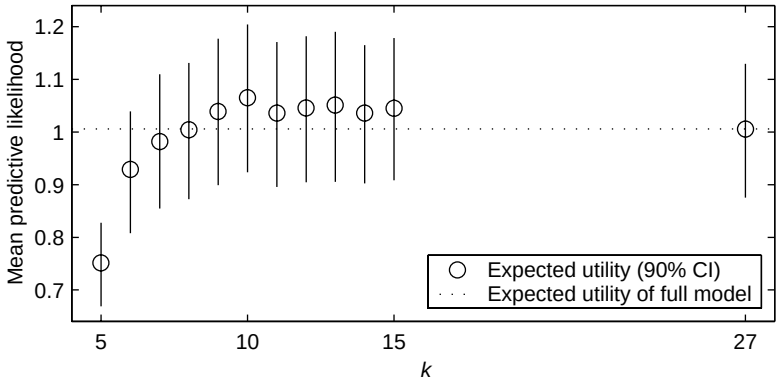

Figure 10: Concrete quality estimation example, predicting the air-% with GP: The expected utilities (mean predictive likelihoods) of the models having the k most probable inputs (see Figure 7). After about nine inputs, adding more inputs does not improve the model performance significantly. To give an impression of the differences in pairwise comparison, there is for example about 90% probability that the nine input model has a higher predictive likelihood than the eight input model.

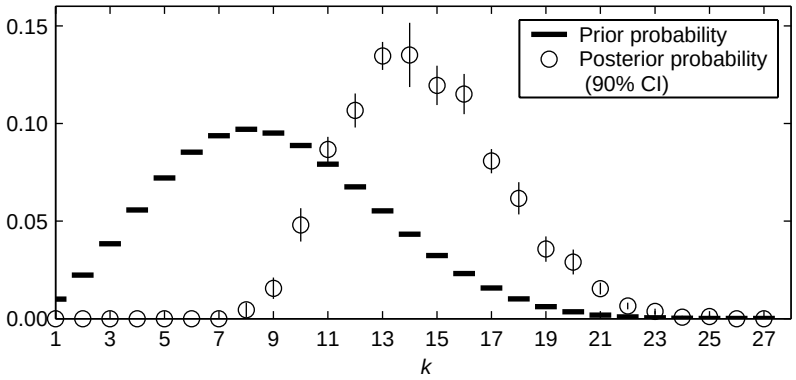

Figure 11: Concrete quality estimation example, predicting the air-% with MLP: The posterior probabilities of the number of inputs with a Beta-bin(27,5,10) prior on the number of inputs. Compare to the results for the GP in Figure 6.

same utility as the best model.

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

Figure 11 shows the posterior probabilities of the number of the inputs with a Beta-bin(27,5,10) prior on the number of inputs. In the case of MLP, larger number of inputs is more probable than in the case of GP (compare to Figure 6). Figure 12 shows the marginal posterior probabilities of the inputs with a Beta-bin(27,5,10) prior on the number of inputs. Most of the inputs have higher posterior probabilities than the mean prior probability (1/3). There is no clear division between more probable inputs and less probable inputs. The nine most probable inputs are same as in the GP case (compare to Figure 7), except that "SC - pore area >300Å" has replaced very similar input "SC - pore area >300-900Å". Figure 13 shows the ARD values of the inputs for the full model. The order of the inputs based on the ARD

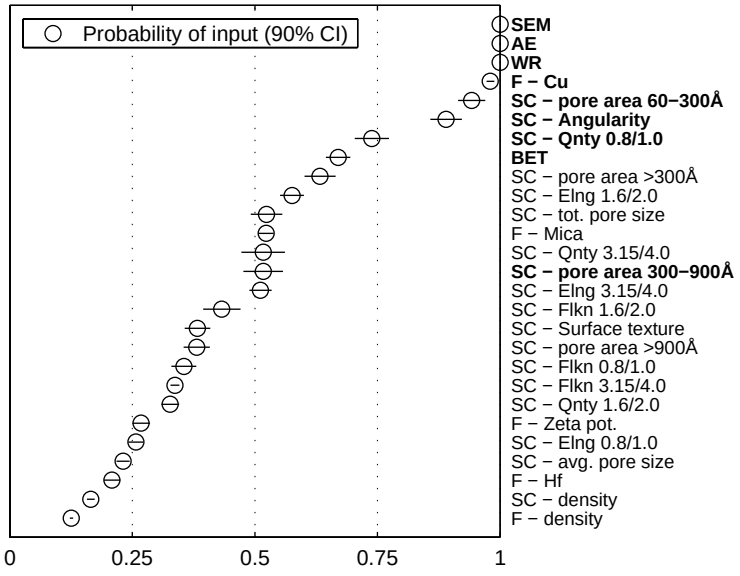

Figure 12: Concrete quality estimation example, predicting the air-% with MLP: The marginal posterior probabilities of inputs with a Beta-bin(27,5,10) prior on the number of inputs. The nine most probable inputs in the GP case are in boldface (see Figure 7).

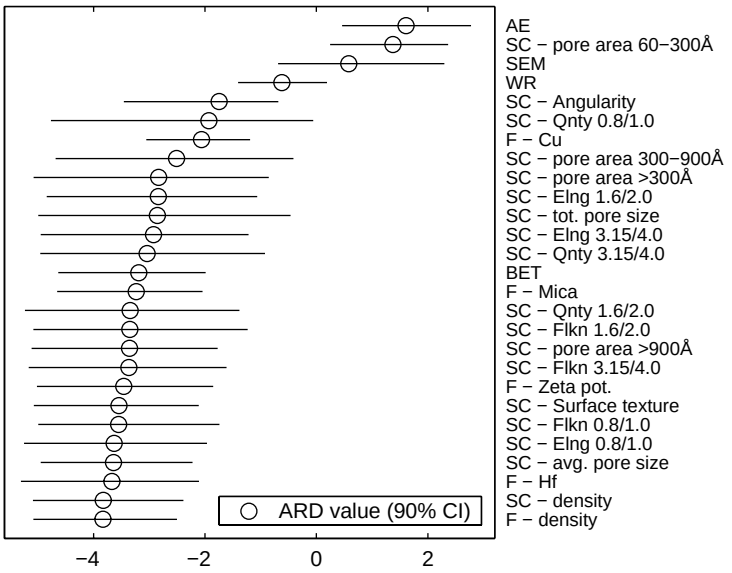

Figure 13: Concrete quality estimation example, predicting the air-% with MLP: The ARD values of the inputs of the full model. Compare to Figure 12.

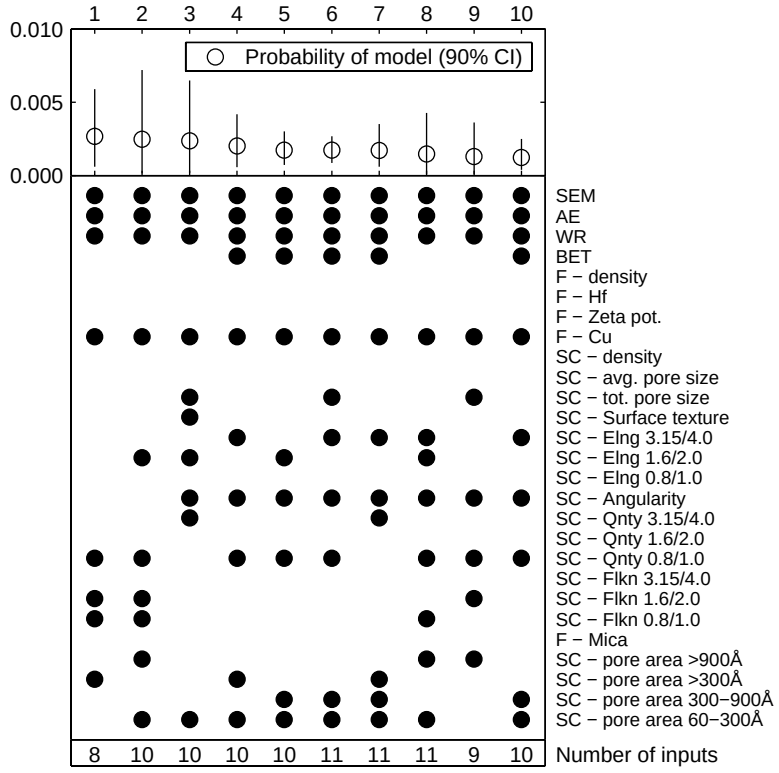

Figure 14: Concrete quality estimation example, predicting the air-% with MLP: The probabilities of the ten most probable models with a Beta-bin(27,5,10) prior on the number of inputs. The top part shows the probabilities of the models, the middle part shows which inputs are in the model, and the bottom part shows the number of inputs in the model. Compare to the results for the GP in Figure 9.

values is clearly different from the order of the inputs based on marginal posterior probabilities (compare to Figure 12). fig:bet_mlp_b_air_pm shows the posterior probabilities of the ten most probable input combinations with a Beta-bin(27,5,10) prior on the number of inputs. There is more variation in the input combinations than in the case of GP and no model is significantly more probable than the others (compare to Figure 9).

Although different inputs would be selected in the case of MLP from the case of GP, the predictive performance (measured with the expected predictive likelihood) was similar for both model types. Figure 15 shows the expected utilities for the full model and the models having the k ($k = 5, \ldots, 15$) most probable inputs. The expected predictive likelihoods are similar for the models having at least about eight most probable inputs and similar to the expected predictive likelihoods of the GP models (Figure 10).

For the bleeding, Figure 16 shows the marginal posterior probabilities of the number of inputs and Figure 17 shows the posterior probabilities of the ten most probable models. About half of the inputs have higher posterior probability than the mean prior probability (1/3). The probability mass has been spread to many inputs and many similar models, due to many correlating inputs. It is less clear than in the case of air-%, which are the most probable inputs and input combinations. However, the most probable models had indistinguishable expected utilities, and thus there were no danger of selecting a bad model. Note how the input *"SC-Qnty 0.8/1.0"* which is included in the most probable model, has

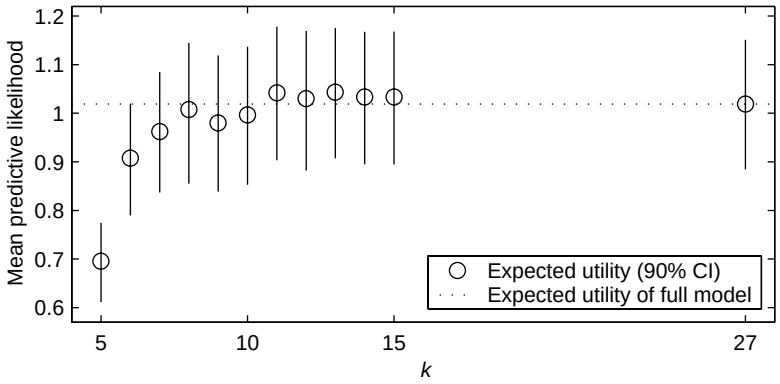

Figure 15: Concrete quality estimation example, predicting the air-% with MLP: The expected utilities of the models having the k most probable inputs (see Figure 12). Compare to results for the GP in Figure 10.

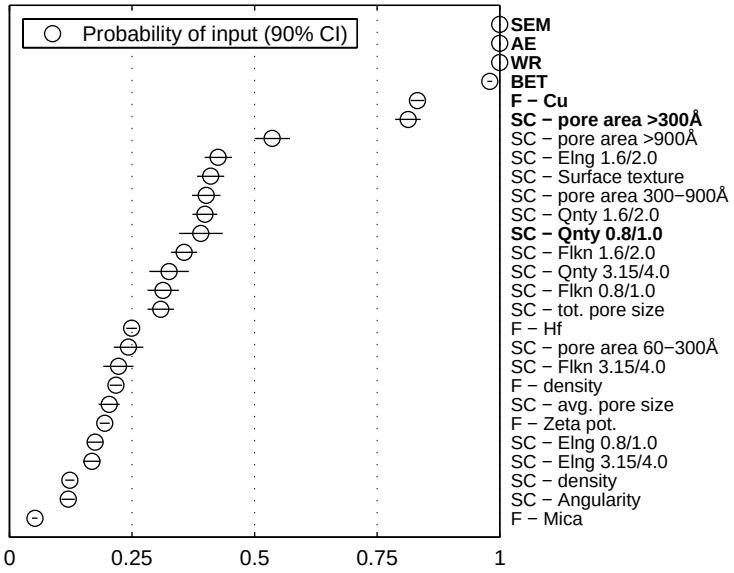

Figure 16: Concrete quality estimation example, predicting bleeding with GP: The marginal posterior probabilities of inputs with Beta-Bin(27, 5, 10) prior on the number of inputs. The inputs in the most probable model are in boldface (see Figure 17).

lower marginal probability than the five other inputs not in that model. This is not peculiar as the five particular inputs correlate strongly with the inputs in the most probable model.

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

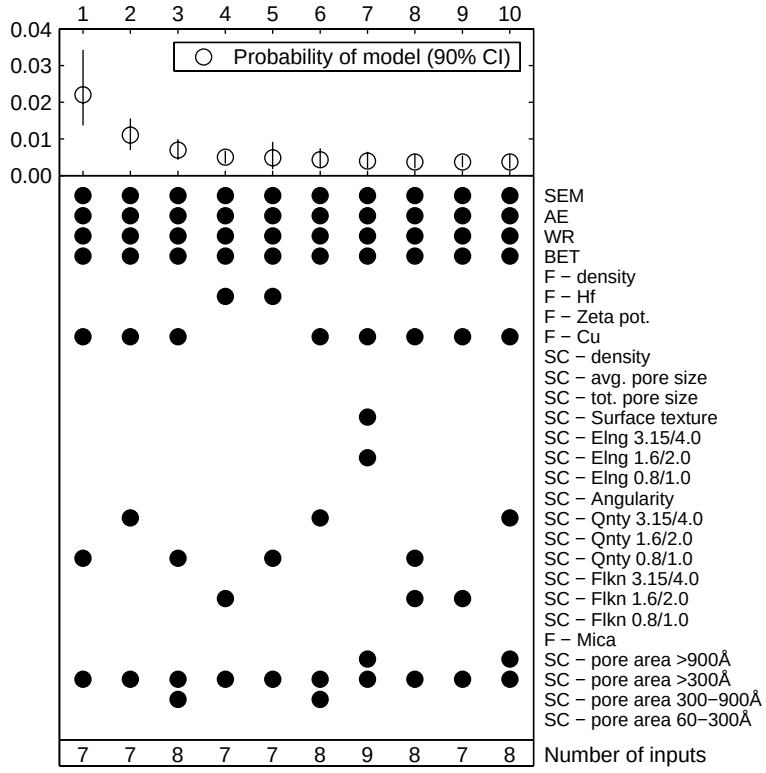

Figure 17: Concrete quality estimation example, predicting bleeding with GP: Probabilities of the ten most probable models with Beta-Bin(27, 5, 10) prior on the number of inputs.

3.6 Real world problem II: Forest scene classification

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

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

The appearance of the tree trunks varies in color and texture due to varying lighting conditions, epiphytes (such as gray or black lichen on white birch), and species dependent variations (such as the Scotch pine, with bark color ranging from dark brown to orange). In the non-tree class the diversity is much larger, containing for example terrain, tree branches and sky. This diversity makes it difficult to choose the optimal features for the classification. We extracted 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. The primary goal was to check if these features contained enough information to produce reasonable classification results (Vehtari et al., 1998). The secondary goal was to reduce the computational burden by reducing the number of features used for the classification.

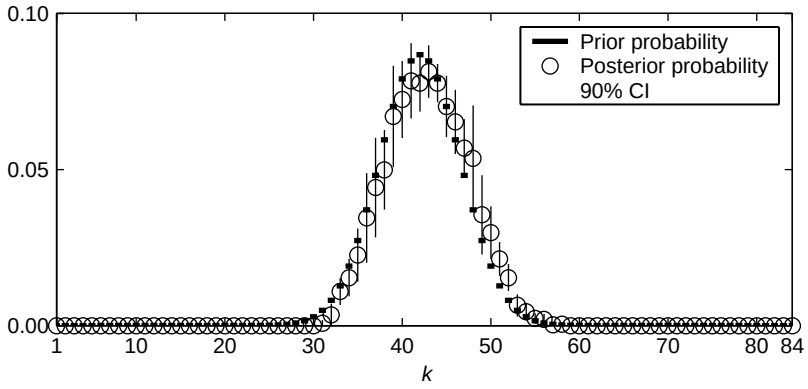

Figure 18: Forest scene classification example: The posterior probabilities of the number of inputs with a uniform prior on the models. The posterior probabilities are similar to prior probabilities and the probability mass is concentrated between 30 and 54 inputs.

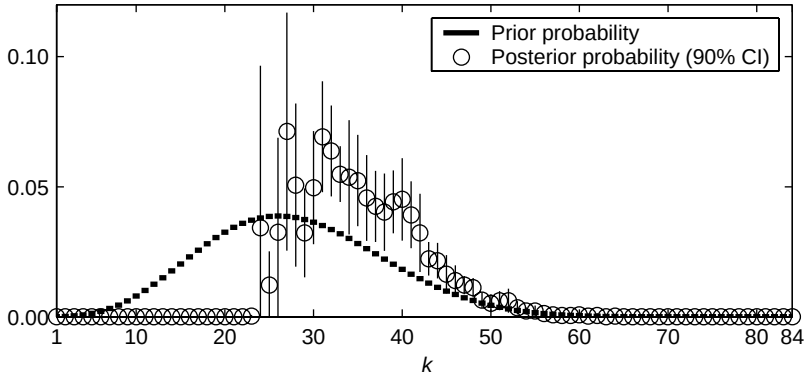

Figure 19: Forest scene classification example: The posterior probabilities of the number of inputs with a Beta-Bin(84,5,15) prior on the number of inputs. The poor between-model convergence can also be noticed from the large uncertainties in the probability estimates seen in this figure (compare to Figure 18).

We used a 20-hidden-unit MLP with the logistic likelihood model. From about $2 \cdot 10^{25}$ possible input combinations, the 4000 saved states included about 3700 and 2500 different input combinations with uniform and Beta priors, respectively. None of the ten independent chains visited any input combination visited by the other chains. Consequently, it was impossible to make good estimates of the probabilities of the input combinations. Instead of trying to obtain an enormous amount of samples, it was possible to choose potentially useful input combinations by using the marginal posterior probabilities of inputs.

Figure 18 shows the posterior probabilities of the number of inputs with equal prior probability for all models. Due to the implicit Binomial prior on the number of inputs (see section 2.3), the probability mass is concentrated between 30 to 54 inputs. Figure 19 shows the posterior probabilities of the number of inputs with a Beta-bin(84,5,15) prior on the number of inputs favoring smaller models. The RJMCMC did not generate samples from models having fewer than 24 inputs (compare to the expected utility results in Figure 21), but this may have been caused by poor between-model convergence when the number of inputs was less than 30. The poor between-model convergence was identified by convergence diagnostics, and it seemed very unlikely that better results could have been obtained in reasonable time.

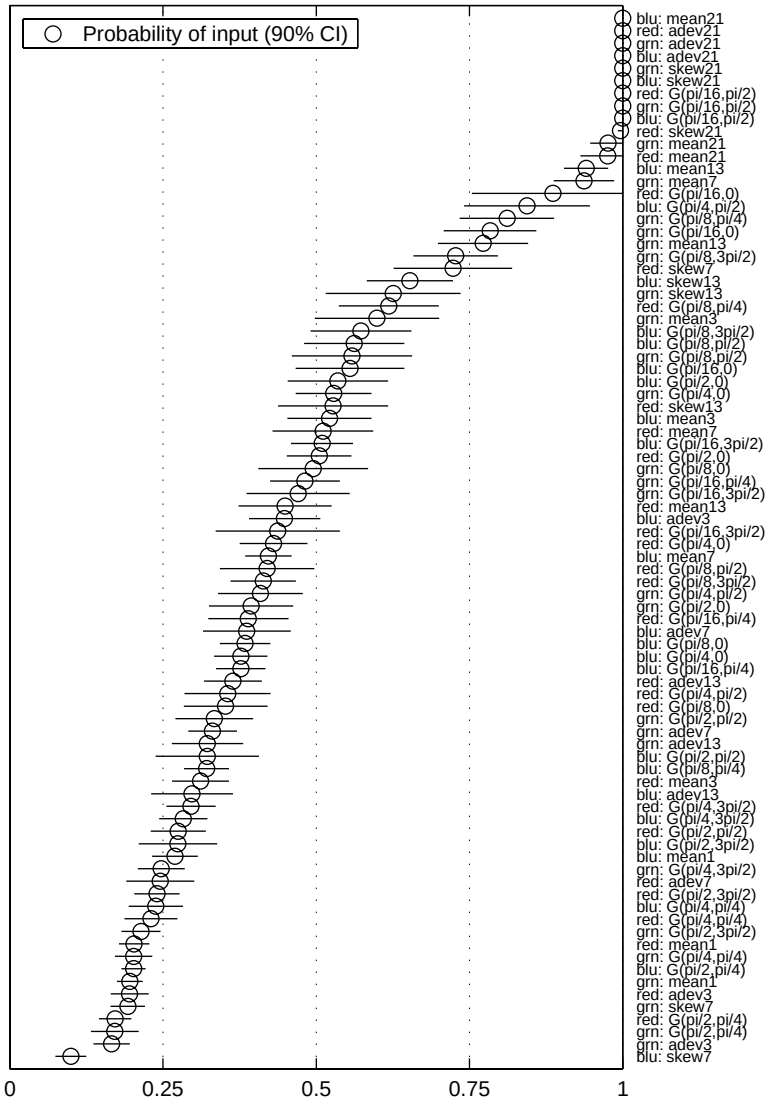

Figure 20: Forest scene classification example: The marginal posterior probabilities of the inputs with a uniform prior on models. These probabilities can be used to estimate the relevance of the inputs.

As the results with a uniform prior on the models had reasonable convergence, it was possible to estimate the relative importance of the inputs using the marginal posterior probabilities of the inputs from that run (Figure 20). Figure 21 shows the comparison of the expected utilities of the models having the k most probable inputs (k between 10 and 40). Reasonable results were achieved also with models having fewer inputs than the smallest model in the RJMCMC. Based on classification accuracy results, just 12 inputs would be sufficient in the planned application. Note that the difference in the performance between the 12 input model and full model is statistically but not practically significant.

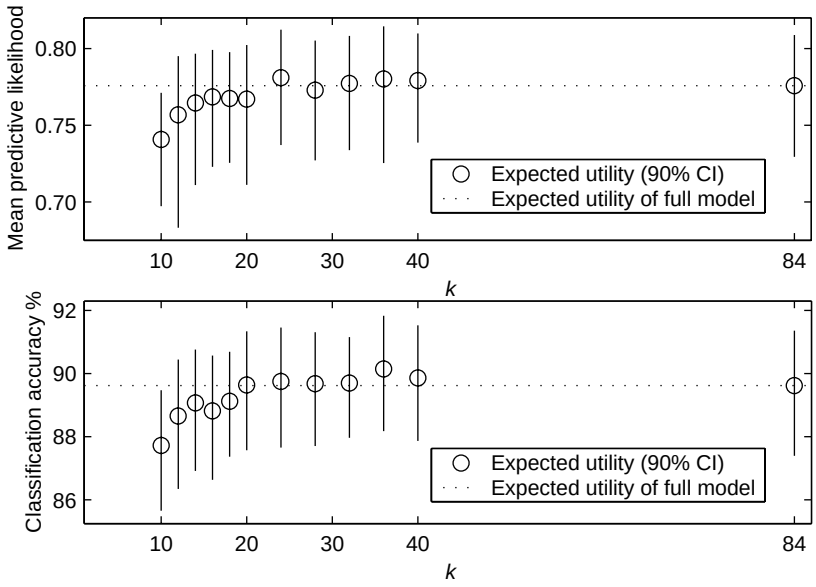

Figure 21: Forest scene classification example: The expected utilities of the models having the k most probable inputs (see Figure 20). The top plot shows the mean predictive likelihood and the bottom plot shows the classification accuracy. The performance starts decreasing when there are fewer than about 20 inputs (compare to the RJMCMC results in Figure 19).

4 Discussion and Conclusion

We have discussed the problem of input variable selection of a complex hierarchical Bayesian model. Our goal was to select a smaller set of input variables in order to make the model more explainable and to reduce the cost of making measurements and the cost of computation. Reducing the number of inputs in the model clearly reduces the cost of making measurements and the cost of computation. If different measurements have different costs we could additionally add these costs to utility function and prefer inputs with low cost of measurement. It is harder to check whether fewer input variables provides model which is easier to analyse.

We motivated our approach using simplicity postulate, which has been criticized because it is not always clear what is simple. For example, using coordinate transforms it may be possible to change non-linear problem to linear problem which may be considered simpler. Also it is not easy to compare how simple different hierarchical Bayesian models are. For example, many random effect models are easy to describe although they contain many parameters and also the effective number of parameters may be very different from the available number of parameters. If the primary goal is to get good predictions and the selection is based on expected utilities, we may get collection of models which have statistically equal predictive power. Then we may choose any one of these without fear of getting poor predictions, and thus we can select the model which we or the application expert feel is simple. In this paper we have considered that models with less input variables are simpler, although the models can still be quite complex as we are using MLP and GP models which are non-linear models capable of handling interactions between inputs. In the concrete problem the application expert successfully used graphical tools to visualize the effects and interactions of the inputs (Järvenpää, 2001).

When considering explainability of the model it is important to remember that we are measuring a probabilistic relationship between inputs and outputs and not a causal one related to the causality

relations in the actual system to be modeled. Thus we can't be sure that all selected variables are causal reason for output. Determination of causal relations is a harder problem (Pearl, 2000). Although causal relation usually implies probabilistic relation, the approach may also exclude input variables which in fact have causal relation to output if data has not enough information about that relation or if our model assumptions are poor.

We proposed to use posterior probabilities obtained via variable dimension MCMC methods to find out potentially useful input combinations and to do the final model choice and assessment using the expected utilities (with any desired utility) computed by using the cross-validation predictive densities. As illustrative examples we used MLP and GP models in one toy problem and in two challenging real world problems. Results show that using posterior probability estimates computed with variable dimension MCMC helps finding useful models in reasonable time and provides insight to models by providing input relevance estimates. Using expected utilities for final input selection reduces the prior sensitivity of the posterior probabilities. By comparing expected utilities of different input combinations we can also make sensitivity checks. Furthermore, expected utility approach provides useful model assessment for the final selected model.

Acknowledgments

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 for helpful comments.

References

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

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.

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

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

Brooks, S. P., Giudici, P., & Philippe, A. (2002). Nonparametric convergence assessment for MCMC model selection. *Journal of Computational and Graphical Statistics*. In press.

Brooks, S. P., Giudici, P., & Roberts, G. O. (2003). Efficient construction of reversible jump MCMC proposal distributions. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*. To appear. Draft available at http://www.statslab.cam.ac.uk/~steve/mypapers/brogr00.ps.

Brown, P. J., Vannucci, M., & Fearn, T. (1998). Multivariate bayesian variable selection and prediction. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 60(3), 627–641.

Brown, P. J., Vannucci, M., & Fearn, T. (2002). Bayes model averaging with selection of regressors. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 64(3), 519–536.

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

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

Chipman, H., George, E. I., & McCulloch, R. E. (2001). Practical implementation of Bayesian model selection (with discussion). In P. Lahiri (ed.), *Model Selection*, vol. 38 of *IMS Lecture Notes – Monograph Series*, (pp. 65–134). Institute of Mathematical Statistics.

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

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

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

Draper, D., & Fouskakis, D. (2000). A case study of stochastic optimization in health policy: Problem formulation and preliminary results. *Journal of Global Optimization*, 18, 399–416.

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.

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

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

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

Han, C., & Carlin, B. P. (2000). MCMC methods for computing Bayes factors: a comparative review. Research Report 2000-001, Division of Biostatistics, University of Minnesota.

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

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

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

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

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

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

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

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. Technical Report 9702, Dept. of Statistics, University of Toronto.

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.

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

Pearl, J. (2000). *Causality: models, reasoning, and inference*. Cambridge University Press.

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

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

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

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

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

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., & van der Linde, A. (2002). Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 64(3), 583–639.

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.

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

Vannucci, M., Brown, P. J., & Fearn, T. (2001). Predictor selection for model averaging. In E. I. George (ed.), *Bayesian Methods with Applications to Science, Policy, and Official Statistics*, (pp. 553–562). International Society for Bayesian Analysis.

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. (2002). Discussion of "Bayesian measures of model complexity and fit" by Spiegelhalter et al. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 64(3), 620.

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.

Vehtari, A., & Lampinen, J. (2002). Bayesian model assessment and comparison using cross-validation predictive densities. *Neural Computation*, 14(10), 2439–2468.

Vehtari, A., & Lampinen, J. (2003). Expected utility estimation via cross-validation. In *Bayesian Statistics 7*. Oxford University Press. In press.