Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007

Search Strategies Guided by the Evidence for the Selection of Basis Functions in Regression

Ignacio Barrio, Enrique Romero, and Lluís Belanche

Abstract— **This work addresses the problem of selecting a subset of basis functions for a model linear in the parameters for regression tasks. Basis functions from a set of candidates are explicitly selected with search methods coming from the feature selection field. Following approximate Bayesian inference, the search is guided by the evidence. The tradeoff between model complexity and computational cost can be controlled by choosing the search strategy. The experimental results show that, under mild assumptions, compact and very competitive models are usually found.**

I. Introduction

In regression tasks we are given a data set of input vectors $\{x_n\}_{n=1}^N$ and corresponding target values $\{t_n\}_{n=1}^N$, where $t_n \in \mathbf{R}$. The objective is to infer a function $y(x)$ that underlies the training data and makes good predictions on unseen input vectors. A very common choice is obtained by a linear model with $m \leq N$ fixed basis functions ϕ_i:

$$y(x; w) = \sum_{i=1}^{m} \omega_i \phi_i(x),$$

where $w = (\omega_1, \omega_2, .., \omega_m)^T$ are the model parameters. Since the model is linear in the parameters, these are easy to estimate and the main problem lies on the selection of the m basis functions (m is unknown a priori) from a *dictionary*. In machine learning, using a dictionary of basis functions centered at the input data usually gives good results [1].

This problem has been mainly tackled in two different ways, according to the *implicit* or *explicit* nature of the selection process. In implicit selection methods, the model with the whole set of basis functions is considered and then the parameters are computed in such a way that many of them become zero. This is the case of Support Vector Machines (SVM) [2], Basis Pursuit (BP) [3], Least Absolute Shrinkage and Selection Operator (LASSO) [4] and Relevance Vector Machines (RVM) [5]. In explicit selection methods a search is carried out guided by the minimization of some cost function. This category includes Matching Pursuits (MP) [6], Orthogonal Least Squares (OLS) [7], Kernel Matching Pursuit (KMP) [8], or some Gaussian process approximations [9], [10], among others. All these methods use *forward selection* as the search strategy.

Explicit selection methods use two criteria: an *objective* (or *cost*) function that conducts the search (*e.g.*, the training set sum-of-squares error) and an *evaluation* function to check model performance, eventually used to stop the process (*e.g.*, the validation set sum-of-squares error). The evaluation is commonly used to avoid overfitting. This duality hinders the use of more powerful search strategies, that would minimize much the first criterion but not necessarily the second one. The choice of a proper objective function is then encouraged if powerful search strategies are to be used.

In a Bayesian setting, under the use of certain priors, there is no need to limit the size of the network to avoid overfitting [11]. However, simpler models are more benefitial for computational reasons. Gaussian processes have been approximated with a subset of regressors [12] and the subset has been selected with forward selection maximizing the marginal likelihood [10], being both the objective and the evaluation function. In the context of linear models, the use of the *evidence* has been suggested to compare different models given that it penalizes complex models and there is (anti)correlation between model evidence and generalization error [13], [14].

In this work we propose an explicit search guided by the evidence for the model. The evidence is both the search objective function and the evaluation function. Several algorithms borrowed from the feature selection field are used as search methods. A fast implementation of the whole process is developed. An experimental study shows that these *Search Strategies Guided by the Evidence* (SSGE) find compact models very competitive with other state-of-the-art techniques such as SVMs and RVMs. More powerful SSGE tend to find more compact models than simpler ones with slightly worse prediction accuracy. By choosing the search strategy the resulting model complexity and the computational cost can be controlled. This control is not possible for SVM and RVM.

The rest of this work is organized as follows. Section II reviews a Bayesian approach for regression with linear models. Section III enumerates some common feature selection search strategies. Section IV presents the SSGE. An experimental study comparing different methods is carried out in Section V and we discuss the results obtained in Section VI. Finally we conclude the paper in Section VII.

II. A Bayesian Approach for Linear Models

We briefly review the noisy interpolation problem and the three levels of inference in a Bayesian framework [14]. The first one considers the posterior distribution over the parameters, the second one adapts the hyperparameters that control the parameters and the third one allows the comparison of different models. We assume the targets to be deviated from the underlying function by independent additive noise

The authors are with the Soft Computing Group, Universitat Politècnica de Catalunya, Barcelona, Spain (email: {ibarrio; eromero; belanche}@lsi.upc.edu).

1-4244-1380-X/07/$25.00 ©2007 IEEE

$t_n = y(x_n; w) + \nu_n$. In linear models the target values are $t = \Phi w + \nu$ where $t = (t_1, t_2, ..t_N)^T$ and $\Phi_{N \times m}$ is a design matrix with entries $\phi_{ij} = \phi_j(x_i)$. If ν is assumed to be zero-mean Gaussian noise with variance σ^2, then the *likelihood* of the parameters is

$$P(t|w, \beta) = \mathcal{N}(\Phi w, \beta^{-1}),$$

where $\beta = 1/\sigma^2$.

A. The first level of inference

Finding the parameters that maximize the likelihood may lead to overfitting. In order to avoid it, the smoothness of $y(x; w)$ is controlled by defining some sort of prior distribution. A common choice is a zero-mean overall Gaussian prior over w,

$$P(w|\alpha) = \mathcal{N}(0, \alpha^{-1}),$$

where α is the inverse variance that measures how smooth $y(x; w)$ is expected to be. Making use of Bayesian inference to find the posterior parameter distribution,

$$P(w|t, \alpha, \beta) = \frac{P(t|w, \beta)P(w|\alpha)}{P(t|\alpha, \beta)}.$$

Since the likelihood and the prior are Gaussian in w, the posterior is also Gaussian:

$$P(w|t, \alpha, \beta) = \mathcal{N}(\mu, \Sigma),$$

where $\Sigma = (\beta\Phi^T\Phi + \alpha I)^{-1}$ and $\mu = \beta\Sigma\Phi^T t$. The parameters w are set to their most probable value μ.

B. The second level of inference

The marginal likelihood is the convolution of Gaussians, $P(t|\alpha, \beta) = \int P(t|w, \beta)P(w|\alpha)dw$, which is also a Gaussian:

$$P(t|\alpha, \beta) = \mathcal{N}(0, \beta^{-1}I + \alpha^{-1}\Phi\Phi^T). \tag{1}$$

In order to find the most suitable values for α and β, again we can make use of Bayes formula: $P(\alpha, \beta|t) = \frac{P(t|\alpha,\beta)P(\alpha,\beta)}{P(t)}$. Assuming we have no prior idea of suitable values for α and β, we consider the prior $P(\alpha, \beta)$ to be uniform on a logarithmic scale $\log\alpha$ and $\log\beta$. Then the most suitable values for α and β are those that maximize $P(t|\alpha, \beta)$. Differentiating (1) and setting the result to zero produces a pair of re-estimation formulae:

$$\alpha_{new} = \frac{\gamma}{||\mu||^2} \qquad \text{and} \qquad \beta_{new} = \frac{N - \gamma}{||t - \Phi\mu||^2}, \tag{2}$$

where $\gamma = m - \alpha\ \mathrm{tr}\Sigma$ is known as the number of well-determined parameters. Equations (2) are dependent on the current value of μ, which is also dependent on α and β. These quantities can be reestimated iteratively, starting from an initial guess for α, β until convergence. We refer to the most probable estimates as $\hat{\alpha}$ and $\hat{\beta}$.

C. The third level of inference

Supposing we have a set of models $\mathcal{H}_i$, containing different subsets of basis functions $\{\phi_i\}$, their posterior probability given the data set is $P(\mathcal{H}_i|t) = \frac{P(t|\mathcal{H}_i)P(\mathcal{H}_i)}{P(t)}$, where $P(\mathcal{H}_i)$ is the prior probability of model $\mathcal{H}_i$. A full Bayesian treatment would use a committee with all the models. A practical approach is to use only the most probable model. Assuming that there is no reason to assign different priors to different models, then we can rank them by evaluating the fully marginalised probability $P(t|\mathcal{H}_i)$, known as the *evidence* for the model.

We can integrate out α and β from (1): $P(t|\mathcal{H}_i) = \int P(t|\alpha, \beta, \mathcal{H}_i)P(\alpha, \beta|\mathcal{H}_i)d\alpha d\beta$, where $P(t|\alpha, \beta, \mathcal{H}_i)$ is the marginal likelihood (1) with the dependency on the model made explicit. This integral has been approximated [14] with a separable Gaussian around $P(t|\hat{\alpha}, \hat{\beta}, \mathcal{H}_i)$:

$$P(t|\mathcal{H}_i) \simeq P(t|\hat{\alpha}, \hat{\beta}, \mathcal{H}_i)P(\hat{\alpha}, \hat{\beta}|\mathcal{H}_i)2\pi\sqrt{\sigma^2_{\log\alpha}\sigma^2_{\log\beta}}, \tag{3}$$

where $\sigma^2_{\log\beta} = 2/(N - \gamma)$ is the variance of the Gaussian approximation for $\log\beta$ and $\sigma^2_{\log\alpha} = 2/\gamma$ is the variance for $\log\alpha$, and they are both found by differentiating (1) twice. This Gaussian approximation holds good for $\gamma \gg 1$ and $N - \gamma \gg 1$ [15]. Again $P(\alpha, \beta|\mathcal{H}_i)$ is considered, as above, a flat prior over $\log\alpha$ and $\log\beta$, so it cancels out when comparing different models. Dropping constant terms, the evidence (or, for convenience, its logarithm) can be approximated by

$$\begin{aligned}\log P(t|\mathcal{H}_i) &\simeq \log P(t|\hat{\alpha}, \hat{\beta}, \mathcal{H}_i) + \frac{1}{2}\log\frac{2}{\gamma} + \frac{1}{2}\log\frac{2}{N - \gamma} \\ &= -\frac{1}{2}\Big[N\log 2\pi + \log|C| + t^T C^{-1}t - \\ &\qquad - \log\frac{2}{\gamma} - \log\frac{2}{N - \gamma}\Big],\end{aligned} \tag{4}$$

with $C = \hat{\beta}^{-1}I + \hat{\alpha}^{-1}\Phi\Phi^T$.

III. Feature Selection Search Strategies

The main objective of feature selection in inductive learning is selecting the most suitable subset from a set of features. The search methods consist of a cost function to direct the search, a policy to decide how to explore new states and an initial state. Some popular methods are:

- **PTA(l, r):** Plus l and Take Away r [16]. At every step, l features are added and then r features are removed (one at a time, always the one that maximizes the objective function). When $l > r$ it is an increasing method, and decreasing for $l < r$. Note forward selection is PTA(1,0).
- **SFFS:** Sequential Forward Floating Selection [17]. At every step, a feature is added and then zero or more features are removed one at a time while the value of the objective function is better than the best value so far with the same number of features. The size does not grow constantly with respect to the number of steps, but in a staggered way.
- **Oscillating(c):** [18] In simplified form, let $s := 1$. Add s features, then remove $2s$ features and add s features

(always one at a time). If the objective function has been maximized, let $s := 1$ and repeat. If not, let $s := s + 1$ and repeat. The algorithm is iterated until $s = c$. The final solution has as many features as the initial one.

IV. Search Strategies Guided by the Evidence

A Bayesian approach suggests the use of the evidence to compare different models. Models with higher evidence usually generalize better and have a sensible number of basis functions. There is the question of which models to compare. We propose to search for a good subset of basis functions from a set of candidates. Unlike other approaches, the basis functions are not restricted to be kernels and the search strategy is not restricted to be forward selection.

Following is the abstract pseudocode for addition/removal of basis functions (an efficient implementation of these operations is presented in the appendix). A *model* is defined by a set of basis functions $\{\phi_i\}$ and the posterior distribution over the parameters (μ, Σ). We use φ_i to denote the candidate basis functions and ϕ_i to denote the basis functions in the current model.

```
AddBestBasisFunction (a model $\mathcal{H}$, $\alpha$, $\beta$, a set of
    candidate basis functions $\{\varphi_i\}$)
 1. for each candidate basis function $\varphi_i$ do
 2.     set $\mathcal{H}'$ the model obtained by adding $\varphi_i$ to $\mathcal{H}$
          and computing the initial value for $\mu'$ and $\Sigma'$
          with equations (section II-A)
 3.     $(\alpha', \beta', \mu', \Sigma')$ := Reestimate$(\alpha, \beta, \mathcal{H}')$
 4.     compute the evidence for $\mathcal{H}'$ with equation (3)
          using $\alpha'$ and $\beta'$
 5. end for
 6. set $\mathcal{H}$ the model obtained by adding to $\mathcal{H}$ the $\varphi_i$
      maximizing the evidence in the previous loop;
      compute the initial value for $\mu$ and $\Sigma$ with
      equations (section II-A)
 7. $(\alpha, \beta, \mu, \Sigma)$ := Reestimate$(\alpha, \beta, \mathcal{H})$
 8. return $(\alpha, \beta, \mathcal{H})$
end AddBestBasisFunction

RemoveWorstBasisFunction (a model $\mathcal{H}$, $\alpha$, $\beta$)
 9. for each basis function $\phi_i$ in $\mathcal{H}$
10.     set $\mathcal{H}'$ the model obtained by removing $\phi_i$ from
          $\mathcal{H}$ and computing the initial value for $\mu'$ and $\Sigma'$
          with equations (section II-A)
11.     $(\alpha', \beta', \mu', \Sigma')$ := Reestimate$(\alpha, \beta, \mathcal{H}')$
12.     compute the evidence for $\mathcal{H}'$ with equation (3)
          using $\alpha'$ and $\beta'$
13. end for
14. set $\mathcal{H}$ the model obtained by removing from $\mathcal{H}$ the
      $\phi_i$ maximizing the evidence in the previous loop;
      compute the initial value for $\mu$ and $\Sigma$ with
      equations (section II-A)
15. $(\alpha, \beta, \mu, \Sigma)$ := Reestimate$(\alpha, \beta, \mathcal{H}')$
16. return $(\alpha, \beta, \mathcal{H})$
end RemoveWorstBasisFunction
```

The call **Reestimate**$(\alpha, \beta, \mathcal{H}')$ iteratively estimates α and β with equations (2) and then the posterior distribution (μ, Σ) with equations (section II-A), until convergence. Initial values of α and β need to be set beforehand. In our experiments, β is set to $(0.1 \times \mathrm{var}(t))^{-1}$ and α is set to 10^{-3}, assuming a broad prior distribution that leaves the weight values fairly unconstrained. We select the first basis function with the largest normalised projection onto the target vector $\|\varphi_i^T t\|^2 / \|\varphi_i\|^2$, following [19]. After that, α and β are reestimated, so their initial values are not directly used for any selection of basis function. Moreover, when trying the addition or removal of a candidate basis function, values of α, β and μ should first be reestimated (steps 3 and 11 in the pseudocode) and the evidence then calculated. In this work, for computational reasons we assume that the previous values of α and β (before addition or removal of a candidate basis function) are rather similar to the posterior ones (after addition or removal). The selection of the best candidate is thus performed with the previous values of α and β, that is, steps 3 and 11 are not implemented. Further, after adding or removing a basis function, α and β should be reestimated (steps 7 and 15). To check convergence, we use (2) for computing trial values α_{new} and β_{new}. Since the evidence is approximated with a Gaussian using $\sigma^2_{\log\alpha}$ and $\sigma^2_{\log\beta}$ (see equation 3), if

$$|\log\beta_{new} - \log\beta| < \epsilon\sqrt{\sigma^2_{\log\beta}} \qquad \text{and}$$
$$|\log\alpha_{new} - \log\alpha| < \epsilon\sqrt{\sigma^2_{\log\alpha}} \tag{5}$$

then we consider convergence is achieved. If not, α, β are set to $\alpha_{new}, \beta_{new}$ and the trial is performed again until condition (5) is met. In the experiments we set $\epsilon = 0.1$. A larger value of ϵ allows faster computations, while a smaller ϵ allows a better convergence of α, β. Sometimes convergence is achieved in the first iteration and the *Reestimate* function does not modify α, β. In this case, the selection of the next basis function can be done more efficiently, as shown in the appendix. The cost of a selection (for addition or removal) is $O(N^2 m)$ when α and β have been modified and $O(N^2)$ when not.

V. Experimental Study

An experimental study comparing different methods is presented. We are interested in assessing the final *evidence* for the models, their *generalization* performance (sum-of-squares error on test set), the model *size* (number of basis functions) and the computational *cost* (number of basis function additions/removals). In addition, we aim to contrast the following conjecture list:

(1) The Bayesian framework deals directly with noise and generates smooth models, therefore the models will hopefully not overfit. We also expect methods like OLS to be outperformed in situations where noise is high. (2) The evidence has been empirically shown to correlate well with generalization performance and with the simplicity of the model [13]. Therefore we expect SSGE to obtain simple models that generalize well. (3) More powerful search strategies like SFFS should obtain higher evidence than simpler ones like forward selection. Therefore, a better generalization and simpler models should be expected for powerful search strategies than for simple ones. (4) Under the use of certain priors, large models usually generalize well [11]. It may be

well possible to obtain similar generalization with SSGE using far fewer basis functions and (5) The SSGEs are competitive with state-of-the-art techniques like SVM or RVM.

The following SSGE methods were tested: PTA(1,0), PTA(2,1), SFFS and Oscillating(5). We compared them to ABF (All the candidate Basis Functions included, maximizing the evidence), OLS, the SVM and a fast greedy implementation of the RVM [19].

DELVE [20] is a collection of data sets and an environment to assess performance of supervised learning methods, allowing for statistically-valid comparisons. We chose the 8-input *pumadyn-8* and *kin-8* data sets. Four versions (prototasks) for each data set are provided: one **f**airly linear with **h**igh noise (*fh*), one **f**airly linear with **m**oderate noise (*fm*), one **n**on-linear with **h**igh noise (*nh*) and one **n**on-linear with **m**oderate noise (*nm*). Each prototask consists of five different tasks, for training set sizes 64, 128, 256, 512 and 1024. Each task has then several task instances, corresponding to particular training and test sets.

We took $\varphi_i(x) = \exp\left(-\sum_{d=1}^{D}\frac{(x_d-c_{id})^2}{r_d^2}\right)$ where $D = 8$ is the input dimension. The centers c_i are the training input vectors. The variables were scaled to fit in $[-1,+1]$. The RBF widths r_d were restricted to be the same for all the methods. In order to set the widths, a model with all the candidate basis functions was considered and conjugate gradients was applied to maximize the marginal likelihood. The RBF widths were jointly optimized with α and β. The model obtained from this first stage is then labelled ABF, and the RBF widths obtained were used for the rest of the methods. The RBF widths were fixed during SSGE learning, so there was no need to integrate them out (like α, β in Section II-C) to compute the evidence for the model.

A stopping criterion is advisable since the highest evidence is usually achieved with a rather small subset of basis functions and much computational cost can be avoided. Assume the current model has m basis functions and the model with the highest evidence up to now has m_h basis functions; if $m_h + k < m$ the process is stopped and the model with m_h basis functions is the solution. A too small k can make the training stop too early, while a k too high will lead to wasteful computation. In this work we set k dynamically to $max(15, 0.3 \times m_h)$, rounded to the closest integer.

The results on DELVE problems are summarized in Table I, which shows the number of tasks where a method performed better than another. We calculated a *t*-test on the test sets averages. Then we considered that a method performed better than another if the *p*-value was lower than 0.05. Each cell shows the number of tasks where the row method performed worse than the column method. Table II shows the mean number of basis functions of the models found with the different algorithms on the tasks with 1024 training data. Table III shows the number of added plus removed basis functions of the different SSGE for the tasks with 1024 training data.

TABLE II
MEAN NUMBER OF BASIS FUNCTIONS OF THE RESULTING MODELS FOR THE TASKS WITH 1024 TRAINING DATA.

Method	pumadyn-8				kin-8			
	fh	**fm**	**nh**	**nm**	**fh**	**fm**	**nh**	**nm**
PTA(1,0)	39	80	59	68	48	71	120	264
PTA(2,1)	11	59	24	34	39	46	103	203
SFFS	2.5	4.5	8.5	13	10	22	76	159
Oscillating(5)	39	80	59	68	48	71	120	264
RVM	3.5	6.5	9.0	13	9.2	24	90	185
SVM	593	726	668	608	708	667	682	836
OLS	11	14	15	26	18	35	49	169

TABLE III
MEAN NUMBER OF ADDED+REMOVED BASIS FUNCTIONS IN THE TRAINING SETS FOR TASKS WITH 1024 TRAINING DATA.

Method	pumadyn-8				kin-8			
	fh	**fm**	**nh**	**nm**	**fh**	**fm**	**nh**	**nm**
PTA(1,0)	56	107	79	88	65	92	155	343
PTA(2,1)	78	238	118	148	162	183	403	791
SFFS	132	344	190	361	330	381	2051	5892
Oscillating(5)	205	308	256	250	224	311	433	738

VI. DISCUSSION

Among the SSGE, SFFS found the solution with the lowest number of basis functions (Table II) and the highest evidence at the expense of the highest cost (Table III). Oscillating(5) had in some cases slightly superior generalization (Table I) than the other SSGE. SFFS had slightly inferior accuracy. The SSGE and the RVM obtained very similar performance than SVM and slightly worse than ABF with a much lower number of basis functions (Table II). The SSGE and the RVM had similar generalization performance with a comparable number of basis functions. ABF, SSGE, RVM and SVM explicitly consider the noise and smoothness in the computation of the parameters, whereas OLS can only control the number of basis functions. We can see that ABF, SSGE, RVM and SVM obtained in many cases better generalization than OLS (Table I). The cost of the RVM and SFFS is very dependent on the task (Table III). In our experiments, PTA(1,0), PTA(2,1) and Oscillating(5) usually required less training time than the RVM, while SFFS required more time.

Conjecture check list

(1) We have seen how the SSGE usually perform better than OLS (see Table I). There will be cases, however, where the assumptions taken by the Bayesian approaches will not hold for the problem at hand and OLS will be superior. (2) The SSGE obtain simple models that generalize well (see Tables I and II). (3) More powerful search strategies like SFFS usually obtain higher evidence and more compact models than simpler ones like forward selection (Table II). We expected more powerful search strategies to generalize better, but that was not true (Table I) because the evidence satisfies a tradeoff between accuracy and simplicity. (4) The SSGE performed slightly worse than a model with all the

TABLE I

SUMMARY OF THE RESULTS. EACH CELL SHOWS THE NUMBER OF TASKS WHERE THE ROW METHOD PERFORMED WORSE THAN THE COLUMN METHOD (p-VALUE LOWER THAN 0.05). THE FIRST FOUR ROWS CORRESPOND TO THE SSGE.

	PTA(1,0)	PTA(2,1)	SFFS	Oscillating(5)	ABF	RVM	SVM	OLS
PTA(1,0)	-	0	0	6	6	3	4	0
PTA(2,1)	3	-	0	3	9	1	4	0
SFFS	9	5	-	11	13	3	6	0
Oscillating(5)	4	0	0	-	7	0	1	0
ABF	1	1	0	2	-	2	2	0
RVM	8	3	1	6	12	-	3	0
SVM	5	2	0	6	5	2	-	0
OLS	21	17	8	19	19	13	11	-

candidate basis functions (ABF) using Bayesian inference (Table I) and (5) The SSGE were competitive (Table I) and produced simpler models than the SVM and comparable to the RVM (Table II). The computational cost of the SVM, however, is lower.

Which SSGE is preferable?

Attaining to our assumptions, since SFFS achieved the highest evidence, it should be the preferred method. However, in practical applications, other aspects become more important than the evidence. Sometimes time is critical; or memory or model size; in other situations, only generalization is important. Using SSGE, these issues can be controlled by selecting the search strategy. Table IV presents the SSGE ordered from top to bottom for the three preferences individually.

TABLE IV

THE SSGE ORDERED FOR EACH PREFERENCE.

−	Training Time	Model Size	Prediction error
↓	PTA(1,0)	SFFS	Oscillating(5)
↓	PTA(2,1)	PTA(2,1)	PTA(1,0)
	Oscillating(5)	PTA(1,0) / Oscil(5)	PTA(2,1)
+	SFFS	PTA(1,0) / Oscil(5)	SFFS

VII. CONCLUSIONS

Following approximate Bayesian inference, we have tackled the regression problem by means of a search guided by the evidence for the model. We have used different search strategies coming from the feature selection field. In the experiments, the Search Strategies Guided by the Evidence (SSGE) produced compact models competitive with state-of-the-art techniques such as the SVM and the RVM. More powerful search strategies found more compact models than simpler ones while maintaining good performance. Unlike SVMs and RVMs, by choosing the search strategy, one can control the model complexity and the computational cost.

Since the model with the highest evidence (found by SFFS) did not obtain the best generalization rate, one could argue that the evidence is not a good measure to optimize. The evidence shows a preference for simpler models, which is why the models with the highest evidence, as found by SFFS, did not have the best performance but were instead much simpler. Nevertheless, this is an appealing property for many practical applications.

APPENDIX: FAST IMPLEMENTATION

Addition of the candidate basis function φ_i makes the C matrix become $C_{+i} = \beta^{-1}I + \alpha^{-1}\Phi\Phi^T + \alpha^{-1}\varphi_i\varphi_i^T = C + \alpha^{-1}\varphi_i\varphi_i^T$, where C_{+i} is C, after the inclusion of basis function i. The determinant and inverse can then be written as

$$|C_{+i}| = |C||1 + \alpha^{-1}\varphi_i^T C^{-1}\varphi_i|,$$
$$C_{+i}^{-1} = C^{-1} - \frac{C^{-1}\varphi_i\varphi_i^T C^{-1}}{\alpha + \varphi_i^T C^{-1}\varphi_i}. \tag{6}$$

Applying them to the marginal likelihood (1), we get

$$\begin{aligned}\log P(t|\alpha,\beta,\mathcal{H}_{+i}) &= \log P(t|\alpha,\beta,\mathcal{H}) + \frac{1}{2}\Big[\log\alpha - \\ &\quad - \log(\alpha + \varphi_i^T C^{-1}\varphi_i) + \frac{(\varphi_i^T C^{-1} t)^2}{\alpha + \varphi_i^T C^{-1}\varphi_i}\Big] \\ &= \log P(t|\alpha,\beta,\mathcal{H}) + \frac{1}{2}\left[\log\alpha - \log(\alpha + S_i) + \frac{Q_i^2}{\alpha + S_i}\right],\end{aligned} \tag{7}$$

with $\mathcal{H}_{+i}$ the model $\mathcal{H}$ with φ_i included and we define $S_i \triangleq \varphi_i^T C^{-1}\varphi_i$ and $Q_i \triangleq \varphi_i^T C^{-1} t$. Using the Woodbury identity lowers the cost of the computation:

$$S_m = \beta\varphi_m^T\varphi_m - \beta^2\varphi_m^T\Phi\Sigma\Phi^T\varphi_m,$$
$$Q_m = \beta\varphi_m^T t - \beta^2\varphi_m^T\Phi\Sigma\Phi^T t.$$

The log-evidence (4) can then be written in an incremental way as

$$\begin{aligned}\log P(t|\mathcal{H}_{+i}) &= \log P(t|\mathcal{H}) + \frac{1}{2}\Big[\log\hat{\alpha} - \log(\hat{\alpha} + S_i) + \\ &+ \frac{Q_i^2}{\hat{\alpha} + S_i} + \log\frac{2}{\gamma_{+i}} + \log\frac{2}{N - \gamma_{+i}} - \log\frac{2}{\gamma} - \log\frac{2}{N-\gamma}\Big],\end{aligned} \tag{8}$$

where γ_{+i} is the number of well determined parameters after adding φ_i. To calculate γ_{+i}, the trace of Σ_{+i} should be computed (see section (II-B)), which is the inverse of a partitioned matrix:

$$\Sigma_{+i} = \begin{pmatrix} \Sigma + \beta^2\Sigma_{ii}\Sigma\Phi^T\varphi_i\varphi_i^T\Phi\Sigma & -\beta\Sigma_{ii}\Sigma\Phi^T\varphi_i \\ -\beta\Sigma_{ii}(\Sigma\Phi^T\varphi_i)^T & \Sigma_{ii} \end{pmatrix}, \tag{9}$$

where $\Sigma_{ii} = (\alpha + S_i)^{-1}$. We can write the trace as $\mathrm{tr}\Sigma_{+i} = \Sigma_{ii} + \beta^2 \Sigma_{ii} R_i + \mathrm{tr}\Sigma$, where we have defined $R_i \triangleq \varphi_i^T \Phi \Sigma \Sigma \Phi^T \varphi_i$. Note that the trace of a product of a column vector by a row vector equals the product of the row vector by the column vector. To add/remove basis functions, it is convenient to maintain and update the values S_m, Q_m and R_m, for each candidate basis function φ_m. When selecting the basis function to add, we need to select the one that most increments the log-evidence (8). This increment can be computed as:

$$\begin{aligned}2(\log P(t|\mathcal{H}_{+i}) - \log P(t|\mathcal{H})) = \log(\hat{\alpha}) - \log(\hat{\alpha} + S_i) + \\ + \frac{Q_i^2}{\alpha + S_i} + \log\frac{\gamma}{\gamma_{+i}} + \log\frac{N-\gamma}{N-\gamma_{+i}},\end{aligned}$$

where $\gamma_{+i} = m + 1 - \alpha \mathrm{tr}\Sigma_{+i}$. If after adding the basis function i, α and β are not modified, then, making use of (9) we can recompute all the S_m, Q_m and R_m incrementally as

$$\begin{aligned}S_{m_{+i}} &= S_m - \Sigma_{ii}(\beta\varphi_m^T e_0)^2, \\ Q_{m_{+i}} &= Q_m - \omega_i(\beta\varphi_m^T e_0), \\ R_{m_{+i}} &= R_m + \varphi_i^T e_2(\Sigma_{ii}\beta\varphi_m^T e_0)^2 + (\varphi_m^T e_1)^2 - (\beta\varphi_m^T e_2)^2,\end{aligned}$$

where $\omega_i = \Sigma_{ii} Q_i$ is the value of the ith parameter after including the basis function, and we define $e_0 \triangleq \varphi_i - \beta\Phi\Sigma\Phi^T\varphi_i$, $e_1 \triangleq \beta\Sigma_{ii}\Phi\Sigma\Phi^T\varphi_i + \beta\Phi\Sigma\Sigma\Phi^T\varphi_i - \Sigma_{ii}\varphi_i$ and $e_2 \triangleq \Phi\Sigma\Sigma\Phi^T\varphi_i$.

For the **removal** of basis function ϕ_i we can rewrite (7):

$$\begin{aligned}\log P(t|\alpha,\beta,\mathcal{H}) &= \log P(t|\alpha,\beta,\mathcal{H}_{-i}) + \frac{1}{2}\Big[\log\alpha - \\ &- \log(\alpha + \phi_i^T C_{-i}^{-1}\phi_i) + \frac{(\phi_i^T C_{-i}^{-1} t)^2}{\alpha + \phi_i^T C_{-i}^{-1}\phi_i}\Big] \\ &= \log P(t|\alpha,\beta,\mathcal{H}_{-i}) + \frac{1}{2}\left[\log\alpha - \log(\alpha + s_i) + \frac{q_i^2}{\alpha + s_i}\right],\end{aligned} \tag{10}$$

where $\mathcal{H}_{-i}$ is the model $\mathcal{H}$ with basis function ϕ_i removed and we have defined $s_i \triangleq \phi_i^T C_{-i}^{-1}\phi_i$ and $q_i \triangleq \phi_i^T C_{-i}^{-1} t$. Applying (6) we can write $s_i = \frac{\alpha S_i}{\alpha - S_i}$, $q_i = \frac{\alpha Q_i}{\alpha - S_i}$. After removing ϕ_i, Σ becomes $\Sigma_{-i} = \Sigma - \frac{1}{\Sigma_{ii}}\Sigma_i\Sigma_i^T$ where we have abused notation, since Σ_{-i} should have one dimension less than Σ (the resulting zeroed ith row and column should be removed). The trace is $\mathrm{tr}\Sigma_{-i} = \mathrm{tr}\Sigma - \frac{1}{\Sigma_{ii}}\Sigma_i^T\Sigma_i$. Substituting s_i and q_i in (10) and extending to the evidence for the model:

$$\begin{aligned}2(\log P(t|\mathcal{H}_{-i}) - \log P(t|\mathcal{H})) = \frac{Q_i^2}{S_i - \alpha} - \log\left(1 - \frac{S_i}{\alpha}\right) + \\ + \log\frac{\gamma}{\gamma_{-i}} + \log\frac{N-\gamma}{N-\gamma_{-i}},\end{aligned}$$

where $\gamma_{-i} = m - 1 - \alpha\mathrm{tr}\Sigma_{-i}$. Again, if α and β are not reestimated after removing ϕ_i, we can write

$$\begin{aligned}S_{m_{-i}} &= S_m + \frac{1}{\Sigma_{ii}}(\beta\Sigma_i^T\Phi^T\varphi_m)^2, \\ Q_{m_{-i}} &= Q_m + \frac{\omega_i}{\Sigma_{ii}}(\beta\Sigma_i^T\Phi^T\varphi_m), \\ R_{m_{-i}} &= R_m - \frac{2}{\Sigma_{ii}}\varphi_m\Phi\Sigma\Sigma_i\Sigma_i^T\Phi\varphi_m + \\ &+ \frac{\Sigma_i^T\Sigma_i}{\Sigma_{ii}}\varphi_m^T\Phi\Sigma_i\Sigma_i^T\Phi^T\varphi_m,\end{aligned}$$

where ω_i is the value of the ith parameter before removing ϕ_i. There is no need to explicitly compute the posterior distribution (μ, Σ in steps 2 and 10 of the pseudocode) in order to compute the evidence.

ACKNOWLEDGMENT

This work has been supported by the Spanish Ministerio de Educación y Ciencia, under projects CGL2004-04702-C02-02 and TIN2006-08114.

REFERENCES

[1] A. J. Smola and B. Schölkopf. On a kernel-based method for pattern recognition, regression, approximation, and operator inversion. *Algorithmica*, 22(1/2):211–231, 1998.

[2] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[3] S. S. Chen, S. L. Donoho, and M. A. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999.

[4] R. J. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc. B*, 58(1):267–288, 1996.

[5] M. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[6] S. G. Mallat and Z. Zhang. Matching Pursuits with Time-Frequency Dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[7] S. Chen, C. F. N. Cowan, and P. M. Grant. Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. *IEEE Transactions on Neural Networks*, 2(2):302–309, 1991.

[8] P. Vincent and Y. Bengio. Kernel matching pursuit. *Machine Learning*, 48(1-3):165–187, 2002.

[9] A. J. Smola and P. L. Bartlett. Sparse greedy Gaussian process regression. In *Advances in Neural Information Processing Systems*, volume 13, pages 619–625, 2001.

[10] J. Quiñonero Candela. *Learning with Uncertainty - Gaussian Processes and Relevance Vector Machines*. PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2004.

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

[12] B. W. Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *J. Roy. Stat. Soc. B*, 47(1):1–52, 1985.

[13] D. J. C. MacKay. A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation*, 4(3):448–472, 1992.

[14] D. J. C. MacKay. Bayesian Interpolation. *Neural Computation*, 4(3):415–447, 1992.

[15] D. J. C. MacKay. Comparison of approximate methods for handling hyperparameters. *Neural Computation*, 11(5):1035–1068, 1999.

[16] J. Kittler. Feature selection and extraction. In Young and Fu, editors, *Handbook of Pattern Recognition and Image Processing*. Academic Press, 1986.

[17] P. Pudil, J. Novovičová, and J. Kittler. Floating Search Methods in Feature Selection. *Pattern Recognition Letters*, 15(11):1119–1125, 1994.

[18] P. Somol and P. Pudil. Oscillating Search Algorithms For Feature Selection. In *Proc. 15th International Conference on Pattern Recognition*, pages 2406–2409, 2000.

[19] M. Tipping and A. Faul. Fast Marginal Likelihood Maximisation for Sparse Bayesian Models. In *Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

[20] C.E. Rasmussen, R.M. Neal, G.E. Hinton, D. van Camp, Z. Ghahramani, M. Revow, Kustra R., and R. Tibshirani. The DELVE Manual, 1996. www.cs.toronto.edu/∼delve/.