Predictive Approaches for Choosing Hyperparameters in Gaussian Processes

S. Sundararajan
Computer Science & Automation
Indian Institute of Science
Bangalore - 560 012
India
sundar@csa.iisc.ernet.in

S.S. Keerthi
Mechanical & Production Engg.
National University of Singapore
10 Kentridge Crescent
Singapore 119260
mpessk@guppy.mpe.nus.edu.sg

Abstract

Gaussian Processes are powerful regression models specified by parameterized mean and covariance functions. Standard approaches to choose these parameters (known by the name hyperparameters) are Maximum Likelihood (ML) and Maximum APosterior (MAP) approaches. In this paper, we propose and investigate predictive approaches based on Geisser's Predictive Sample Reuse (PSR) methodology and the related Stone's Cross Validation (CV) methodology. More specifically, we derive results for Geisser's surrogate Predictive Probability (GPP), Geisser's Predictive mean square Error (GPE) and the standard CV error and make a comparative study. Within an approximation we arrive at the Generalized Cross Validation (GCV) and establish its relationship with the GPP and GPE approaches. These approaches are tested on a number of problems. Experimental results show that these approaches are strongly competitive to the existing approaches.

1 Introduction

Gaussian Processes (GPs) are powerful regression models that have gained popularity recently, though they have appeared in different forms in the literature for years. They can be used for classification also; see MacKay (1997), Neal (1997), Rasmussen (1996) and Williams and Rasmussen (1996). Here, we restrict ourselves to regression problems. GPs are specified using parametric forms for the mean and covariance functions (Williams and Rasmussen (1996)). We assume that the process is zero mean. Let $\mathbf{Z}_N = \{\mathbf{X}_N, \mathbf{t}_N\}$ where $\mathbf{X}_N = \{\mathbf{x}(i) : \ i = 1, \ldots, N\}$ and $\mathbf{t}_N = \{t(i) : \ i = 1, \ldots, N\}$. Here, $t(i)$ represents the observed output corresponding to the input vector $\mathbf{x}(i)$. Then, the likelihood (MacKay (1997)) is given by

$$p(\mathbf{t}_N|\mathbf{X}_N, \boldsymbol{\theta}) \ = \ \frac{exp(-\mathbf{t}_N^T\mathbf{C}_N^{-1}\mathbf{t}_N)}{(2\pi)^{\frac{N}{2}}|\mathbf{C}_N|^{\frac{1}{2}}} \tag{1}$$

where $\mathbf{C}_N \ = \ \bar{\mathbf{C}}_N \ + \ \sigma^2\mathbf{I}_N$, $\bar{\mathbf{C}}_N$ is the covariance matrix with $(i,j)^{th}$ element $[\bar{\mathbf{C}}_N]_{i,j} \ = \ \bar{C}(\mathbf{x}(i), \mathbf{x}(j); \bar{\boldsymbol{\theta}})$, and $\bar{C}(\cdot; \bar{\boldsymbol{\theta}})$ denotes the covariance function parametrized by $\bar{\boldsymbol{\theta}}$. Note that $\boldsymbol{\theta} = (\bar{\boldsymbol{\theta}},\sigma^2)$ is the set of hyperparameters. Then, the predictive distribution of the output $y(N+1)$ for a test case $\mathbf{x}(N+1)$ is also Gaussian with mean and variance

$$\hat{y}(N+1) \ = \ \mathbf{k}_{N+1}^T\mathbf{C}_N^{-1}\mathbf{t}_N \tag{2}$$

and

$$\sigma^2_{y(N+1)} \ = \ b_{N+1} \ - \ \mathbf{k}_{N+1}^T\mathbf{C}_N^{-1}\mathbf{k}_{N+1} \tag{3}$$

where $b_{N+1} = C(\mathbf{x}(N+1), \mathbf{x}(N+1); \boldsymbol{\theta})$ and $\mathbf{k}_{N+1}$ is an $N \times 1$ vector with i^{th} element given by $C(\mathbf{x}(N+1), \mathbf{x}(i); \boldsymbol{\theta})$. We need to specify the covariance function $C(\cdot; \boldsymbol{\theta})$. Williams and Rasmussen (1996) found the following covariance function to work well in practice.

$$\bar{C}(\mathbf{x}(i), \mathbf{x}(j); \bar{\boldsymbol{\theta}}) \ = \ a_0 \ + \ a_1 \sum_{p=1}^{M} x_p(i)x_p(j) \ + \ v_0 \, exp(-\frac{1}{2}\sum_{p=1}^{M} w_p \, (x_p(i) \ - \ x_p(j))^2) \tag{4}$$

where $x_p(i)$ is p^{th} component of i^{th} input vector $\mathbf{x}(i)$. The parameters w_1, w_2,..., w_M are known by the name Automatic Relevance Determiniation (ARD) hyperparameters (Rasmussen (1996)) since their values influence the relevance of the inputs in determining the output and are determined automatically using the training data. Note that all the parameters are positive and it is convenient to use logarithmic scale. Hence, $\boldsymbol{\theta}$ is given by $log(a_0, a_1, v_0, w_1, \ldots, w_M, \sigma^2)$. Then, the question is: how do we handle $\boldsymbol{\theta}$? Sophisticated techniques like Hybrid Monte Carlo (HMC) methods (Rasmussen (1996) and Neal (1997)) are available which can numerically integrate over the hyperparameters to make predictions. Alternately, we can estimate $\boldsymbol{\theta}$ from the training data. We restrict to the latter approach here. In the classical approach, $\boldsymbol{\theta}$ is assumed to be deterministic but unknown and the ML estimate $\boldsymbol{\theta_{ML}}$ is found by maximizing the likelihood (1). In the Bayesian approach, $\boldsymbol{\theta}$ is assumed to be random and a prior $p(\boldsymbol{\theta})$ is specified. Then, the MAP estimate $\boldsymbol{\theta}_{MP}$ is found by maximizing $p(\mathbf{t}_N|\mathbf{X}_N, \boldsymbol{\theta})p(\boldsymbol{\theta})$ with the motivation that the predictive distribution $p(y(N+1)|\mathbf{x}(N+1), \mathbf{Z}_N)$ can be approximated as $p(y(N+1)|\mathbf{x}(N+1), \mathbf{Z}_N, \boldsymbol{\theta}_{MP})$.

We proposed and investigated different predictive approaches based on Geisser's Predictive Sample Reuse (PSR) methodology (Geisser (1975)) and the related Stone's cross-validation (CV) methodology (Stone (1974)) to estimate the hyperparameters (from the training data) in linear models (Sundararajan (1999a)) and artificial neural networks (Sundararajan and Keerthi (1999a) and Sundararajan (1999b)). In this paper we investigate these approaches to choose the hyperparameters in GPs. An earlier version of this paper has been accepted for NIPS'99 (Sundararajan and Keerthi (1999b)). Simulation results indicate that the proposed approaches are strongly competitive to the existing approaches.

2 Predictive Approaches for Choosing Hyperparameters

Geisser (1975) proposed PSR methodology that can be applied for both model selection and parameter estimation problems. This methodology is closely related to the CV methodology independently presented by Stone (1974). Based on the PSR methodology, Geisser and Eddy (1979) proposed to maximize $\prod_{i=1}^{N} p(t(i)|\mathbf{Z}_N^{(i)}, \mathbf{x}(i), M_j)$ (known as Geisser's surrogate Predictive Probability (GPP)) by synthesizing Bayesian and PSR methodology in the context of (parameterized) model selection. Here, we propose to maximize GPP $= \prod_{i=1}^{N} p(t(i)|\mathbf{Z}_N^{(i)}, \mathbf{x}(i), \boldsymbol{\theta})$ to estimate $\boldsymbol{\theta}$, where $\mathbf{Z}_N^{(i)}$ is obtained from $\mathbf{Z}_N$ by removing the i^{th} sample. Note that $p(t(i)|\mathbf{Z}_N^{(i)}, \mathbf{x}(i), \boldsymbol{\theta})$ is nothing but the predictive distribution $p(y(i)|\mathbf{Z}_N^{(i)}, \mathbf{x}(i), \boldsymbol{\theta})$ evaluated at $y(i) = t(i)$. Since maximizing GPP is equivalent to minimizing its negative logarithm we define the GPP function as

$$G(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{i=1}^{N} log(p(t(i)|\mathbf{Z}_N^{(i)}, \mathbf{x}(i), \boldsymbol{\theta})) \tag{5}$$

Also, we define Geisser's Predictive mean square Error (GPE) as

$$G_E(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} E((y(i) - t(i))^2) \tag{6}$$

where the expectation operation is defined with respect to $p(y(i)|\mathbf{Z}_N^{(i)}, \mathbf{x}(i), \boldsymbol{\theta})$ and propose to estimate $\boldsymbol{\theta}$ by minimizing GPE. One can show that (see Burman (1989) and Efron and Gong (1983)) our proposal to choosing the hyperparameters by maximizing GPP is equivalent to choosing the hyperparameters such that the implied conditional density function (model) is having maximum similarity with the true conditional density function. On the other hand, finding the hyperparameters that minimize GPE is equivalent to choosing the hyperparameters that minimize the predictive mean square error. Next, let us define the CV error as

$$H(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} (t(i) - \hat{y}(i))^2 \tag{7}$$

where $\hat{y}(i)$ is the mean of the conditional predictive distribution $p(y(i)|\mathbf{Z}_N^{(i)}, \mathbf{x}(i), \boldsymbol{\theta})$. Then, (6) can be written as

$$G_E(\boldsymbol{\theta}) = H(\boldsymbol{\theta}) + \frac{1}{N} \sum_{i=1}^{N} \sigma_{y(i)}^2 \tag{8}$$

where $\sigma_{y(i)}^2$ is the variance of $p(y(i)|\mathbf{Z}_N^{(i)}, \mathbf{x}(i), \boldsymbol{\theta})$. Thus, while the CV error minimizes the deviation from the predictive mean, GPE takes the predictive variance also into account.

Expressions for the predictive measures and their gradients. From (2) and (3) we see that $p(y(i)|\mathbf{Z}_N^{(i)}, \mathbf{x}(i), \boldsymbol{\theta})$ is Gaussian with mean $\hat{y}(i)$ and variance $\sigma^2_{y(i)}$. Further, the summand in (5) is evaluated at $y(i) = t(i)$. Therefore, to evaluate the various predictive measures, we only need to find expressions for the error $t(i) - \hat{y}(i)$ and $\sigma^2_{y(i)}$; these can be evaluated efficiently. These expressions are derived in the appendix. Then, $G(\boldsymbol{\theta})$ and its gradient can be computed efficiently using the following result.

Theorem 1 *The objective function $G(\boldsymbol{\theta})$ under the Gaussian Process model is given by*

$$G(\boldsymbol{\theta}) = \frac{1}{2N}\sum_{i=1}^{N}\frac{q_N^2(i)}{\bar{c}_{ii}} - \frac{1}{2N}\sum_{i=1}^{N} log\bar{c}_{ii} + \frac{1}{2}log2\pi \tag{9}$$

where $\bar{c}_{ii}$ denotes the i^{th} diagonal entry of $\mathbf{C}_N^{-1}$ and $q_N(i)$ denotes the i^{th} element of $\mathbf{q}_N = \mathbf{C}_N^{-1}\mathbf{t}_N$. Its gradient is given by

$$\frac{\partial G(\boldsymbol{\theta})}{\partial \theta_j} = \frac{1}{2N}\sum_{i=1}^{N}\Big(1 + \frac{q_N^2(i)}{\bar{c}_{ii}}\Big)\Big(\frac{s_{j,i}}{\bar{c}_{ii}}\Big) + \frac{1}{N}\sum_{i=1}^{N} q_N(i)\Big(\frac{r_j(i)}{\bar{c}_{ii}}\Big) \tag{10}$$

where $s_{j,i} = \bar{\mathbf{c}}_i^T \frac{\partial \mathbf{C}_N}{\partial \theta_j}\bar{\mathbf{c}}_i$, $\mathbf{r}_j = -\mathbf{C}_N^{-1}\frac{\partial \mathbf{C}_N}{\partial \theta_j}\mathbf{C}_N^{-1}\mathbf{t}_N$ and $\mathbf{q}_N = \mathbf{C}_N^{-1}\mathbf{t}_N$. Here, $\bar{\mathbf{c}}_i$ denotes the i^{th} column of the matrix $\mathbf{C}_N^{-1}$.

We will give meaningful interpretation to the different terms shortly. Similarly, using the following result we can compute $H(\boldsymbol{\theta})$ and its gradient efficiently.

Theorem 2 *The CV function $H(\boldsymbol{\theta})$ under the Gaussian model is given by*

$$H(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\Big(\frac{q_N(i)}{\bar{c}_{ii}}\Big)^2 \tag{11}$$

and its gradient is given by

$$\frac{\partial H(\boldsymbol{\theta})}{\partial \theta_j} = \frac{1}{N}\sum_{i=1}^{N}\Big(\frac{2}{\bar{c}_{ii}}\Big)\Big(\frac{q_N(i)r_j(i)}{\bar{c}_{ii}} + \Big(\frac{q_N^2(i)}{\bar{c}_{ii}}\Big)\Big(\frac{s_{j,i}}{\bar{c}_{ii}}\Big)\Big) \tag{12}$$

where $s_{j,i}, \mathbf{r}_j, q_N(i)$ and $\bar{c}_{ii}$ are as defined in theorem 1.

Next, from lemma 1 (see the appendix) we can obtain the expression for $G_E(\boldsymbol{\theta})$ from (8) and (11) and the gradient can be written as

$$\frac{\partial G_E(\boldsymbol{\theta})}{\partial \theta_j} \;=\; \frac{\partial H(\boldsymbol{\theta})}{\partial \theta_j} \;+\; \frac{1}{N}\sum_{i=1}^{N}\Big(\frac{1}{\bar{c}_{ii}}\Big)^2 \bar{\mathbf{c}}_i^T \frac{\partial \mathbf{C}_N}{\partial \theta_j}\bar{\mathbf{c}}_i \tag{13}$$

The expressions derived above can be used together with powerful gradient based optimization methods to determine the hyperparameters.

Interpretations. More insight can be obtained from reparameterizing the covariance function as follows.

$$C(\mathbf{x}(i),\mathbf{x}(j);\boldsymbol{\theta}) = \sigma^2\Big(\bar{a}_0+\bar{a}_1\sum_{p=1}^{M}\mathbf{x}_p(i)\mathbf{x}_p(j)+\bar{v}_0 exp(-\frac{1}{2}\sum_{p=1}^{M} w_p(\mathbf{x}_p(i)-\mathbf{x}_p(j))^2)+\delta_{i,j}\Big) \tag{14}$$

where $a_0 = \sigma^2\bar{a}_0$, $a_1 = \sigma^2\bar{a}_1$, $v_0 = \sigma^2\bar{v}_0$. Let us define $P(\mathbf{x}(i),\mathbf{x}(j);\boldsymbol{\theta}) = \frac{1}{\sigma^2}C(\mathbf{x}(i),\mathbf{x}(j);\boldsymbol{\theta})$. Then $\mathbf{P}_N^{-1} \;=\; \sigma^2\, \mathbf{C}_N^{-1}$. Therefore, $\bar{c}_{i,j} \;=\; \frac{\bar{p}_{i,j}}{\sigma^2}$ where $\bar{c}_{i,j}$, $\bar{p}_{i,j}$ denote the $(i,j)^{th}$ element of the matrices $\mathbf{C}_N^{-1}$ and $\mathbf{P}_N^{-1}$ respectively. Then, with $\bar{\mathbf{q}}_N \;=\; \mathbf{P}_N^{-1}\mathbf{t}_N$ we can rewrite (9) as

$$G(\boldsymbol{\theta}) \;=\; \frac{1}{2N\sigma^2}\sum_{i=1}^{N}\frac{\bar{q}_N^2(i)}{\bar{p}_{ii}} \;-\; \frac{1}{2N}\sum_{i=1}^{N} log\bar{p}_{ii} \;+\; \frac{1}{2}log2\pi\sigma^2 \tag{15}$$

Now, by setting the derivative of (15) with respect to σ^2 to zero, we can infer the noise level as

$$\hat{\sigma}^2 \;=\; \frac{1}{N}\sum_{i=1}^{N}\frac{\bar{q}_N^2(i)}{\bar{p}_{ii}} \tag{16}$$

Similarly, the CV error (7) can be rewritten as

$$H(\boldsymbol{\theta}) \;=\; \frac{1}{N}\sum_{i=1}^{N}\frac{\bar{q}_N^2(i)}{\bar{p}_{ii}^2} \tag{17}$$

Note that $H(\boldsymbol{\theta})$ is dependent only on the ratio of the hyperparameters (i.e., on $\bar{a}_0, \bar{a}_1, \bar{v}_0$) apart from the ARD parameters. Therefore, we cannot infer the noise level uniquely. However, we can estimate the ARD parameters and the ratios $\bar{a}_0, \bar{a}_1, \bar{v}_0$. Once we have estimated these parameters, then we can use (16) to estimate the noise level. Next, we note that the noise level preferred by the GPE criterion is zero. To see this, first let us rewrite (8) under reparameterization as

$$G_E(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\frac{\bar{q}_N^2(i)}{\bar{p}_{ii}^2} + \sigma^2\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\bar{p}_{ii}} \tag{18}$$

Since $\bar{q}_N(i)$ and $\bar{p}_{ii}$ are independent of σ^2, it follows that the GPE prefers zero as the noise level, which is not true. Therefore, this approach can be applied when, either the noise level is known or a good estimate of it is available.

GCV, Approximate GPP, GPE and Their Relationships. We can use an approximation similar to the one used with the standard CV error function (Breiman (1995)) of a linear model that result in the Generalized Cross Validation (GCV) (Craven and Wahba (1979)), for the GP model also. More specifically, let $\tilde{p} = \frac{1}{N}\sum_{i=1}^{N}\bar{p}_{ii}$ and use the approximation $\bar{p}_{ii} \approx \tilde{p},\ \forall\, i$. With this approximation, let us denote $G(\boldsymbol{\theta})$, $G_E(\boldsymbol{\theta})$, $H(\boldsymbol{\theta})$ and $\hat{\sigma}^2$ as $\tilde{G}(\boldsymbol{\theta})$, $\tilde{G}_E(\boldsymbol{\theta})$, $\tilde{H}(\boldsymbol{\theta})$ and $\tilde{\sigma}^2$ respectively and refer $\tilde{G}(\boldsymbol{\theta})$ and $\tilde{G}_E(\boldsymbol{\theta})$ as the approximate GPP and GPE functions respectively. GCV (which is same as $\tilde{H}(\boldsymbol{\theta})$) is given by

$$V(\boldsymbol{\theta}) = \frac{\sum_{i=1}^{N}\bar{q}_N^2(i)}{N\tilde{p}^2} \tag{19}$$

and $\tilde{\sigma}^2 = \tilde{p}V(\boldsymbol{\theta})$. Unlike in the case of linear model, the computational complexity of the GCV measure is not reduced and remains the same as the CV error in its present form here. But, this approximation sheds light on the relationships between the measures. First, the derivatives of $\tilde{G}(\boldsymbol{\theta})$ and $V(\boldsymbol{\theta})$ in the reparameterized form are related through

$$\frac{\partial \tilde{G}(\boldsymbol{\theta})}{\partial \theta_j} = \Big(\frac{\partial V(\boldsymbol{\theta})}{\partial \theta_j}\Big)\Big(\frac{1}{2V(\boldsymbol{\theta})}\Big) \tag{20}$$

Noting that $V(\boldsymbol{\theta}) > 0$, we see that if $\frac{\partial \tilde{G}(\boldsymbol{\theta})}{\partial \theta_j} = 0, \forall j$, then $\frac{\partial V(\boldsymbol{\theta})}{\partial \theta_j} = 0, \forall j$. This implies that the first order necessary condition at the extremum (i.e., minimum or maximum) point is satisfied at the same point for both the approximate GPP and GCV functions. To prove that both these functions attain the minimum or maximum at the same points we need to study the properties of the Hessian matrices, which is difficult. However, based on our results on linear models (Sundararajan (1999a)), we expect that the minimizer/maximizer of these two objective functions to be same. Next, consider the approximate GPE and GCV. We have already seen that the noise level preferred by GPE is zero. Therefore, to use GPE either the noise level must be known or at least a good estimate of it is available. Alternately, we can substitute $\tilde{\sigma}^2$ in place of σ^2 in $\tilde{G}_E(\boldsymbol{\theta})$ and then estimate the hyperparameters in the reparameterized form. This yields $\tilde{G}_E(\boldsymbol{\theta}) = 2\,V(\boldsymbol{\theta})$. As a consequence the solutions obtained by minimizing the GCV and approximate GPE are exactly the same.

3 Simulation Results

We have carried out simulation on four different data sets. We considered MacKay's robot arm problem and its modified version introduced by Neal (1996). We used the same data set as MacKay (2-inputs and 2-outputs), with 200 examples in the training set and 200 in the test set. This data set is referred as data set 1 in Table 1. Next, to evaluate the ability of the predictive approaches in estimating the ARD parameters, we carried out simulation on the robot arm data with 6 inputs (Neal's version), denoted as data set 2 in Table 1. This data set was generated by adding four further inputs, two of which were copies of the two inputs corrupted by additive zero mean Gaussian noise of standard deviation 0.02 and two further irrelevant Gaussian noise inputs with zero mean and unit variance (Williams and Rasmussen (1996)). The performance measures chosen were average of Test Set Error (TSE) (normalized by true noise level of 0.0025) and average of negative logarithm of predictive probability (NLPP) (computed from Gaussian density function with (2) and (3)). Friedman's (Friedman (1991)) data sets 3 and 4 were based on the problem of predicting impedance and phase respectively from four parameters of an electrical circuit. Training sets of four different sizes (50, 100, 150, 200) and with a signal-to-noise ratio of about 3:1 were replicated 100 times and for each training set (at each sample size N), scaled integral squared error ($ISE = \frac{\int_D (y(\mathbf{x}) - \hat{y}(\mathbf{x}))^2 d\mathbf{x}}{var_D\, y(\mathbf{x})}$) and NLPP were computed using 5000 data points randomly generated from a uniform distribution over D (Friedman (1991)).

The approaches considered here are the existing ML, MAP approaches and the GPP, CV, GCV and GPE approaches developed here. We have seen that the noise level cannot be estimated directly from the GPE and CV approaches. One possible approach is to use the noise level estimate obtained from any other approach (say, for example, the one obtained from GPP). In the simulation we have used the noise level estimate generated randomly with 10% error (that is, Gaussian distributed with true noise level NL_T as the mean and standard deviation of 0.03 NL_T). These approaches are denoted as GPE_K and CV_K in the tables. Comparison of these two approaches helps in understanding the important role played by the predictive variance. In the case of CV approach, we have also estimated the hyperparameters in the reparameterized form. This approach is denoted as CV_{UK} in the tables. Note that eventhough in the CV approach the ratio of the hyperparameters (other than ARD) with the noise level is important, in practical implementations the solutions obtained from CV_K and CV_{UK} may be different. In the case of MAP, we used the same prior given in Rasmussen (1996).

For all methods, conjugate gradient (CG) algorithm (Rasmussen (1996)) was used to optimize the hyperparameters. The termination criterion (relative function error) with a tolerance of 10^{-7} was used, but with a constraint on the maximum number of CG iterations set to 100. In the case of robot arm data sets, the algorithm was run with ten different initial conditions and the best solution (chosen from respective best objective function value) is reported. The optimization was carried out separately for the two outputs and the results reported are the average TSE, NLPP. In the case of Friedman's data sets, the optimization algorithm was run with three different initial conditions and the best solution was picked up. When $N = 200$, the optimization algorithm was run with only one initial condition. For all the data sets, both the inputs and outputs were normalized to zero mean and unit variance.

Table 1 gives the results on the robot arm data sets. We see that the performance (both TSE and NLPP) of the predictive approaches is better than ML and MAP approaches for both the data sets. In the case of data set 2, we observed that like ML and MAP methods, all the predictive approaches rightly identified the irrelvant inputs. The performance of the GPE approach is the best on the robot arm data and demonstrates

	$Data\ Set:1$		$Data\ Set:2$	
	TSE	NLPP	TSE	NLPP
ML	1.126	-1.512	1.131	-1.512
MP	1.131	-1.511	1.181	-1.489
GPP	1.115	-1.524	1.116	-1.516
CV_{UK}	1.112	-1.518	1.146	-1.514
GCV	1.113	-1.516	1.119	-1.509
CV_K	1.115	-1.520	1.147	-1.514
GPE_K	1.111	-1.524	1.112	-1.524

Table 1: Results on robot arm data sets. Average of normalized test set error and negative logarithm of predictive probability for various methods.

	N = 50	N = 100	N = 150	N = 200
ML	0.43(0.20)	0.19(0.09)	0.13(0.06)	0.10(0.04)
MAP	0.42(0.15)	0.22(0.09)	0.16(0.06)	0.12(0.04)
GPP	0.47(0.28)	0.20(0.12)	0.13(0.08)	0.10(0.04)
CV_{UK}	0.55(0.34)	0.22(0.12)	0.13(0.08)	0.10(0.05)
GCV	-	0.30(0.24)	0.14(0.08)	0.10(0.05)
CV_K	0.54(0.36)	0.21(0.11)	0.13(0.07)	0.09(0.04)
GPE_K	0.35(0.17)	0.15(0.07)	0.11(0.05)	0.08(0.03)

Table 2: Friedman's Data Set:3. Integral Squared Error (ISE) for different training sample sizes. Mean and standard deviation (in brackets) of ISE obtained from 100 different training data sets.

the usefulness of this approach when a good noise level estimate is available. Next, note that the performance of GCV is very close to the standard CV indicating the validity of the approximation introduced.

In the case of Friedman's data set 3, (see Tables 2, 3) the important observation is that the performance (both TSE and NLPP) of the predictive approaches (GPP, CV_K and CV_{UK}) is poor at low sample size ($N = 50$) and it improves very well as the sample size increases. Note that the performance of the predictive approaches is better compared to the ML and MAP methods starting from $N = 100$ onwards. Also, note that eventhough the TSE performance of the ML approach is better or same as that of GPP, the NLPP performance of the GPP approach is better from $N = 100$ onwards. This is because in NLPP, the normalized TSE (normalized by the predictive variance) in compromise with the logarithm of the predictive variance is important. This phenomenon can be observed for several other cases as well. Next, the performance of the CV_{UK} method is almost same or slightly inferior to the CV_K method. The close performance indicates that there

	N = 50	N = 100	N = 150	N = 200
ML	7.24(0.73)	6.71(0.17)	6.58(0.10)	6.49(0.08)
MAP	7.18(0.29)	6.78(0.16)	6.65(0.10)	6.56(0.09)
GPP	7.29(0.59)	6.65(0.17)	6.52(0.13)	6.44(0.10)
CV_{UK}	7.27(0.63)	6.67(0.17)	6.50(0.13)	6.44(0.09)
GCV	-	6.83(0.36)	6.54(0.14)	6.45(0.11)
CV_K	7.28(0.90)	6.68(0.20)	6.50(0.13)	6.41(0.09)
GPE_K	7.10(0.46)	6.60(0.20)	6.46(0.13)	6.37(0.07)

Table 3: Friedman's Data Set:3. Average of Negative Logarithm of Predictive Probability (NLPP) of the test set for different training sample sizes. Mean and standard deviation (in brackets) of NLPP obtained from 100 different training data sets.

	N = 50	N = 100	N = 150	N = 200
ML	0.26(0.07)	0.16(0.02)	0.14(0.02)	0.11(0.01)
MAP	0.25(0.06)	0.16(0.02)	0.13(0.02)	0.11(0.01)
GPP	0.33(0.13)	0.20(0.07)	0.15(0.03)	0.12(0.02)
CV_{UK}	0.42(0.28)	0.21(0.08)	0.16(0.04)	0.13(0.03)
GCV	-	0.24(0.11)	0.16(0.05)	0.13(0.04)
CV_K	0.38(0.24)	0.21(0.07)	0.16(0.04)	0.13(0.02)
GPE_K	0.28(0.08)	0.18(0.04)	0.14(0.02)	0.12(0.02)

Table 4: Friedman's Data Set:4. Integral Squared Error (ISE) for different training sample sizes. Mean and standard deviation (in brackets) of ISE obtained from 100 different training data sets.

	N = 50	N = 100	N = 150	N = 200
ML	1.05(0.17)	0.82(0.05)	0.74(0.05)	0.68(0.04)
MAP	1.01(0.11)	0.82(0.12)	0.74(0.05)	0.69(0.04)
GPP	1.25(0.49)	0.86(0.10)	0.76(0.06)	0.70(0.05)
CV_{UK}	1.36(0.47)	0.91(0.22)	0.79(0.08)	0.70(0.05)
GCV	-	0.93(0.24)	0.78(0.08)	0.72(0.10)
CV_K	1.25(0.30)	0.93(0.20)	0.78(0.25)	0.66(0.08)
GPE_K	1.20(0.27)	0.85(0.16)	0.72(0.09)	0.63(0.06)

Table 5: Friedman's Data Set:4. Average of Negative Logarithm of Predictive Probability (NLPP) of the test set for different training sample sizes. Mean and standard deviation (in brackets) of NLPP obtained from 100 different training data sets.

is really no need to know the noise level to use the CV approach.

In the case of GCV, we have not reported the results for $N = 50$. This is because we faced the problem of noise level going to zero for several training sets. We believe that this behavior is because of the invalidity of the approximation at low sample size. This belief is based on the following observation. We observed that when $N = 100$, for 10 training data sets (in the case of data set 4) and 3 training data sets (in the case of data set 3) had similar behavior. The results reported are performances excluding these cases. When $N = 150$ we observed this behavior for 3 training data sets (in the case of data set 4) and none (in the case of data set 3). When $N = 200$, we did not observe any such behavior. So, as one would expect the approximation becomes better and better as the number of samples increases. This can be seen by comparing the performance of GCV with that of CV_{UK}.

Among all the approaches GPE gives the best performance. This could be due to the use of fairly accurate noise level estimates. But, note that the performance is much better compared to that of the CV_K method. This behavior, we expect is because of the important role played by the predictive variance. Also, the performance, even at low sample size ($N = 50$), is quite good. Next, it is clear that the MAP method gives the best performance at low sample size. This behavior, we believe is because, in this case the prior plays an important role and hence is very useful.

In the case of Friedman's data set 4, (see Tables 4, 5) the ML and MAP approaches perform better compared to the predictive approaches except GPE_K. As mentioned earlier, the performance of predictive methods improves as the number of samples increases and is very close to the ML and MAP methods when $N = 150$, $N = 200$. On comparing the performances of CV_K, GPE_K, we once again see the role and importance of the predictive variance. Also, note that unlike the previous data set, the performance of GPE_K is inferior to the ML and MAP approaches at low sample sizes and improves over these approaches (see NLPP) as N increases.

4 Conclusion

The predictive approaches estimate the predictive performance of a given model from the training samples alone. Since the quality of the estimate will be better as the number of examples increases, the performance of these approaches improves very well with increase in the number of samples. Further, the knowledge of noise level helps in improving the quality of this estimate. Typically, when the number of samples is sufficiently large, we find that the predictive approaches perform better than the ML and MAP approaches. In particular, we find that the MAP approach is the best when the number of samples is very low. Simulation results indicate that the number of samples required to get good estimates (for the predictive approaches to perform better than the existing approaches) will be dependent on the problem. The *sufficient* number of samples can be as low as 100 as evident from our results on Friedman's data set 3. It is not clear how one can find the number of samples required to take advantage of these methods before actually trying out on a given data set. The comparison with the existing approaches indicate that the predictive approaches developed here are strongly competitive. Among the predictive approaches we recommend GPP approach to choose the hyperparameters. When the noise level is known or a good estimate of it is available, GPE approach is an alternate useful approach. Finally, the overall cost for computing the function and the gradient (for all the predictive approaches) is $O(MN^3)$; on the other hand, the cost is $O(N^3)$ for the ML and MAP approaches. However, the cost for making prediction is same for all the approaches.

Appendix

Let $\mathbf{E}_{i,N} = (\mathbf{e_1} \quad \mathbf{e}_2 \quad \ldots \quad \mathbf{e}_{i-1} \quad \mathbf{e}_N \quad \mathbf{e}_{i+1} \quad \ldots \quad \mathbf{e}_{N-1} \quad \mathbf{e}_i)$, where $\mathbf{e}_j$ is the i^{th} column vector of the identity matrix $\mathbf{I}_N$. Then, $\mathbf{E}_{i,N}$ is symmetric, orthogonal. See Golub and Van Loan (1985). Now, let $\mathbf{C}_N = (\mathbf{c_1} \quad \mathbf{c}_2 \quad \ldots \quad \mathbf{c}_N)$ and $\mathbf{C}_{i,N} = \mathbf{E}_{i,N}\mathbf{C}_N\mathbf{E}_{i,N}$. Then,

$$\mathbf{C}_{i,N} = \begin{pmatrix} \mathbf{C}_N^{(i)} & \mathbf{c}_i^{(i)} \\ [\mathbf{c}_i^{(i)}]^T & c_{ii} \end{pmatrix} \tag{21}$$

where $\mathbf{c}_i^{(i)} = \mathbf{c}_i - \{c_{ii}\}$, that is, $\mathbf{c}_i^{(i)}$ is an $N-1\times 1$ vector obtained from removing the i^{th} element c_{ii} from $\mathbf{c}_i$ and the matrix $\mathbf{C}_N^{(i)}$ is obtained from $\mathbf{C}_N$ by removing the i^{th} row and column.

Lemma 1 *The variance* $\sigma^2_{y(i)}$ *is given by* $\frac{1}{\bar{c}_{ii}}$ *and the error term* $t(i) - \hat{y}(i)$ *is given by* $\frac{q_N(i)}{\bar{c}_{ii}}$. *Here,* $q_N(i)$ *is the* i^{th} *element of the vector* $\mathbf{q}_N = \mathbf{C}_N^{-1}\mathbf{t}_N$ *and* $\bar{c}_{ii}$ *is the* i^{th} *diagonal element of* $\mathbf{C}_N^{-1}$. *Further,* $\frac{\partial \bar{c}_{ii}}{\partial \theta_j} = -\bar{\mathbf{c}}_i^T \frac{\partial \mathbf{C}_N}{\partial \theta_j}\bar{\mathbf{c}}_i$ *and* $\frac{\partial \mathbf{q}_N}{\partial \theta_j} = -\mathbf{C}_N^{-1}\frac{\partial \mathbf{C}_N}{\partial \theta_j}\mathbf{C}_N^{-1}\mathbf{t}_N$.

Proof: From (2) and (3), we have

$$\hat{y}(i) = [\mathbf{c}_i^{(i)}]^T[\mathbf{C}_N^{(i)}]^{-1}\mathbf{t}_N^{(i)} \tag{22}$$

$$\sigma^2_{y(i)} = c_{ii} - [\mathbf{c}_i^{(i)}]^T[\mathbf{C}_N^{(i)}]^{-1}\mathbf{c}_i^{(i)} \tag{23}$$

The results follow from simplifying these expressions. First, note that

$$\mathbf{C}_{i,N}^{-1} = \mathbf{E}_{i,N}\mathbf{C}_N^{-1}\mathbf{E}_{i,N} \tag{24}$$

since, $\mathbf{E}_{i,N}$ is both orthogonal and symmetric. Therefore, $\mathbf{C}_{i,N}^{-1}$ is nothing but the matrix obtained by interchanging the i^{th} row and column of $\mathbf{C}_N^{-1}$ with N^{th} row and column respectively. Now, if we write $\mathbf{C}_N^{-1} = (\ \bar{\mathbf{c}}_1 \quad \bar{\mathbf{c}}_2 \quad \ldots \quad \bar{\mathbf{c}}_N\ )$, then

$$\mathbf{C}_{i,N}^{-1} = \begin{pmatrix} \bar{\mathbf{C}}_N^{(i)} & \bar{\mathbf{c}}_i^{(i)} \\ [\bar{\mathbf{c}}_i^{(i)}]^T & \bar{c}_{ii} \end{pmatrix} \tag{25}$$

where $\bar{\mathbf{C}}_N^{(i)}$ is obtained from $\mathbf{C}_N^{-1}$ by removing the i^{th} column and row and $\begin{pmatrix} [\bar{\mathbf{c}}_i^{(i)}]^T & \bar{c}_{ii} \end{pmatrix}^T = \mathbf{E}_{i,N}\bar{\mathbf{c}}_i$. Note that $\bar{c}_{ii}$ is the i^{th} diagonal entry of $\mathbf{C}_N^{-1}$. Next, using a result on inverse of a partitioned matrix (see problem 2.7, page 33 in Rao (1991)) on $\mathbf{C}_{i,N}$ (see (21)), we find

$$\mathbf{C}_{i,N}^{-1} = \begin{pmatrix} [\mathbf{C}_N^{(i)}]^{-1} + \frac{\mathbf{d}_i\mathbf{d}_i^T}{c_{ii} - \mathbf{d}_i^T\mathbf{c}_i^{(i)}} & -\frac{\mathbf{d}_i}{c_{ii} - \mathbf{d}_i^T\mathbf{c}_i^{(i)}} \\ -\frac{\mathbf{d}_i^T}{c_{ii} - \mathbf{d}_i^T\mathbf{c}_i^{(i)}} & \frac{1}{c_{ii} - \mathbf{d}_i^T\mathbf{c}_i^{(i)}} \end{pmatrix} \tag{26}$$

where $\mathbf{d}_i = [\mathbf{C}_N^{(i)}]^{-1}\mathbf{c}_i^{(i)}$. On comparing the $(N,N)^{th}$ element of (25) and (26) , we see that $\frac{1}{c_{ii} - \mathbf{d}_i^T\mathbf{c}_i^{(i)}} = \bar{c}_{ii}$. Then, from (23) it follows that $\sigma^2_{y(i)} = \frac{1}{\bar{c}_{ii}}$.

From (26), we see that the N^{th} element of $\mathbf{C}_{i,N}^{-1}\begin{pmatrix}\mathbf{t}_N^{(i)}\\ t(i)\end{pmatrix}$ is nothing but $(t(i) - [\mathbf{c}_i^{(i)}]^T[\mathbf{C}_N^{(i)}]^{-1}\mathbf{t}_N^{(i)})\bar{c}_{ii}$. But from (24),

$$\mathbf{C}_{i,N}^{-1}\begin{pmatrix}\mathbf{t}_N^{(i)}\\ t(i)\end{pmatrix} = \mathbf{E}_{i,N}\mathbf{C}_N^{-1}\mathbf{E}_{i,N}\mathbf{E}_{i,N}\mathbf{t}_N = \mathbf{E}_{i,N}\mathbf{C}_N^{-1}\mathbf{t}_N = \mathbf{E}_{i,N}\mathbf{q}_N$$

with N^{th} element as $q_N(i)$. Therefore, from (22) it follows that $t(i) - \hat{y}(i) = \frac{q_N(i)}{\bar{c}_{ii}}$. To find the derivatives, we use the result $\frac{\partial \mathbf{C}_N^{-1}}{\partial \theta_j} = -\mathbf{C}_N^{-1}\frac{\partial \mathbf{C}_N}{\partial \theta_j}\mathbf{C}_N^{-1}$. This completes the proof.

References

Breiman, L. (1995) Better subset regression using the non-negative garrote. *Technometrics*, **37(4)**, 373-384.

Burman, P. (1989) A Comparative Study of Ordinary Cross-Validation and the Repeated Learning Testing Methods. *Biometrika*, **76**, **3**, 503-14.

Craven, P. and Wahba, G. (1979) Smoothing noisy data with spline functions. *Numer. Math.*, **31**, 377-403.

Efron, B., Gong, G. (1983) A Leisurely Look at the Bootstrap, the Jackknife and Cross-Validation. *The American Statistician*, **37**, **1**, 36-48.

Friedman, J.H., (1991) Multivariate Adaptive Regression Splines, *Ann. of Stat.*, **19**, 1-141.

Geisser, S., (1975) The Predictive Sample Reuse Method with Applications, *Journal of the American Statistical Association*, **70**, 320-328.

Geisser, S., and Eddy, W.F., (1979) A Predictive Approach to Model Selection, *Journal of the American Statistical Association*, **74**, 153-160.

Golub, G.H. and Van Loan, C.F. (1985) *Matrix Computations*, Baltimore: The John Hopkins University Press.

MacKay, D.J.C. (1997) Gaussian Processes - *A replacement for neural networks ?*, Available in Postscript via *URL http://www.wol.ra.phy.cam.ac.uk/mackay/.*

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

Neal, R.M. (1997) Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification. Tech. Rep. No. 9702, Dept. of Statistics, University of Toronto.

Rasmussen, C. (1996) *Evaluation of Gaussian Processes and other Methods for Non-Linear Regression*, Ph.D. Thesis, Dept. of Computer Science, University of Toronto.

Rao, C.R. (1991) *Linear Statistical Inference and Its Applications*, New Delhi: Wiley Eastern Limited.

Stone, M. (1974) Cross-Validatory Choice and Assessment of Statistical Predictions (with discussion), *J. R. Statist. Soc.* B, **36**, 111-147.

Sundararajan, S. (1999a) Predictive approaches for choosing hyperparameters and estimating noise level in linear regression problems. Submitted to *J. R. Statist. Soc.* B and is under revision.

Sundararajan, S. (1999b) *Predictive approaches for choosing hyperparameters in linear and nonlinear regression problems.* Ph.D. Thesis, Dept. of Computer Science and Automation, Indian Insitute of Science, Bangalore, India.

Sundararajan, S. and Keerthi, S.S. (1999a) Predictive approaches for choosing hyperparameters in artifical neural networks. To be Submitted to *IEEE Trans. on Neural Networks.*

Sundararajan, S. and Keerthi, S.S. (1999b) Predictive approaches for choosing hyperparameters in Gaussian Processes. Accepted for the conference on *Advances in Neural Information Processing Systems (NIPS 12).*

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