Variational Learning of Inducing Variables in Sparse Gaussian Processes

Michalis K. Titsias
School of Computer Science,
University of Manchester, UK
mtitsias@cs.man.ac.uk

Abstract

Sparse Gaussian process methods that use inducing variables require the selection of the inducing inputs and the kernel hyperparameters. We introduce a variational formulation for sparse approximations that jointly infers the inducing inputs and the kernel hyperparameters by maximizing a lower bound of the true log marginal likelihood. The key property of this formulation is that the inducing inputs are defined to be variational parameters which are selected by minimizing the Kullback-Leibler divergence between the variational distribution and the exact posterior distribution over the latent function values. We apply this technique to regression and we compare it with other approaches in the literature.

1 INTRODUCTION

The application of Gaussian process (GP) models is intractable for large datasets because the time complexity scales as $O(n^3)$ and the storage as $O(n^2)$ where n is the number of training examples. To overcome this limitation, many approximate or sparse methods have been proposed in the literature (Williams and Seeger, 2001; Smola and Bartlett, 2001; Csato and Opper, 2002; Lawrence et al., 2002; Seeger et al., 2003; Schwaighofer and Tresp, 2003; Snelson and Ghahramani, 2006; Quiñonero-Candela and Rasmussen, 2005). These methods construct an approximation based on a small set of m support or inducing variables that allow the reduction of the time complexity from $O(n^3)$ to $O(nm^2)$. They mainly differ in the strategies they use to select the inducing inputs which are typically selected from the training or test examples. Snelson and Ghahramani (2006) allow the inducing variables to be considered as auxiliary pseudo-inputs that are inferred along with kernel hyperparameters using continuous optimization.

Approximate marginal likelihoods are appropriate objective functions for model selection in sparse GP models. Existing state-of-the-art methods (Snelson and Ghahramani, 2006; Seeger et al., 2003) derive such approximations by modifying the GP prior (Quiñonero-Candela and Rasmussen, 2005) and then computing the marginal likelihood of the modified model. This approach turns the inducing inputs into additional kernel hyperparameters. While this can increase flexibility when we fit the data, it can also lead to overfitting when we optimize with respect to all unknown hyperparameters. Furthermore, fitting a modified model is not so rigorous approximation procedure since there is no distance between the exact and the modified model that is minimized.

In this paper we introduce a variational method that jointly selects the inducing inputs and the hyperparameters by maximizing a lower bound to the exact marginal likelihood. The important difference between this formulation and previous methods is that here the inducing inputs are defined to be variational parameters which are selected by minimizing the Kullback-Leibler (KL) divergence between a variational GP and the true posterior GP. This allows i) to avoid overfitting and ii) to rigorously approximate the exact GP model by minimizing a distance between the sparse model and the exact one. The selection of the inducing inputs and hyperparameters is achieved either by applying continuous optimization over all unknown quantities or by using a variational EM algorithm where at the E step we greedily select the inducing inputs from the training data and at the M step we update the hyperparameters. In contrast to previous greedy ap-

Appearing in Proceedings of the 12^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5.

proaches, e.g. (Seeger et al., 2003), our scheme monotonically increases the optimized objective function.

We apply the variational method to regression with additive Gaussian noise and we compare its performance to training schemes based on the projected process marginal likelihood (Seeger et al., 2003; Csato and Opper, 2002) and the sparse pseudo-inputs marginal likelihood (Snelson and Ghahramani, 2006).

Our method is most closely related to the variational sparse GP method described in (Csato and Opper, 2002; Seeger, 2003) that is applied to GP classification (Seeger, 2003). The main difference between our formulation and these techniques is that we maximize a variational lower bound in order to select the inducing inputs, while these methods use variational bounds for estimating only the kernel hyperparameters.

2 SPARSE GP REGRESSION

A GP is a set of random variables $\{\mathbf{f}(\mathbf{x})|\mathbf{x} \in \mathcal{X}\}$ for which any finite subset follows a Gaussian distribution. To describe a GP, we only need to specify the mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$. The covariance function typically depends on a set of hyperparameters $\boldsymbol{\theta}$. A GP can be used as a prior over a real-valued function $f(\mathbf{x})$. This prior can be combined with data to give a posterior over the function.

Suppose we have a training dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ of n noisy realizations of some unobserved or latent function so that each scalar y_i is obtained by adding Gaussian noise to $f(\mathbf{x})$ at input $\mathbf{x}_i$, i.e. $y_i = f_i + \epsilon_i$, where $\epsilon_i \sim N(0, \sigma^2)$ and $f_i = f(\mathbf{x}_i)$. Let X denote all training inputs, $\mathbf{y}$ all outputs and $\mathbf{f}$ the corresponding training latent function values. The joint probability model is $p(\mathbf{y}, \mathbf{f}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f})$ where $p(\mathbf{y}|\mathbf{f})$ is the likelihood and $p(\mathbf{f})$ the GP prior. The data induce a posterior GP which is specified by a posterior mean function and a posterior covariance function:

$$m_{\mathbf{y}}(\mathbf{x}) = K_{\mathbf{x}n}(\sigma^2 I + K_{nn})^{-1}\mathbf{y}, \qquad (1)$$

$$k_{\mathbf{y}}(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - K_{\mathbf{x}n}(\sigma^2 I + K_{nn})^{-1}K_{n\mathbf{x}'}.$$

Here, K_{nn} is the $n \times n$ covariance matrix on the training inputs, $K_{\mathbf{x}n}$ is n-dimensional row vector of kernel function values between $\mathbf{x}$ and the training inputs and $K_{n\mathbf{x}} = K_{\mathbf{x}n}^T$. Any query related to the posterior GP can be answered by the above mean and covariance functions. For instance, the Gaussian posterior distribution $p(\mathbf{f}|\mathbf{y})$ on the training latent variables $\mathbf{f}$ is computed by evaluating eq. (1) at the inputs X. Similarly the prediction of the output $y_* = f_* + \epsilon_*$ at some unseen input $\mathbf{x}_*$ is described by $p(y_*|\mathbf{y}) = N(y_*|m_{\mathbf{y}}(\mathbf{x}_*), k_{\mathbf{y}}(\mathbf{x}_*, \mathbf{x}_*) + \sigma^2)$. The posterior GP depends on the values of the hyperparameters $(\boldsymbol{\theta}, \sigma^2)$ which can be estimated by maximizing the log marginal likelihood given by

$$\log p(\mathbf{y}) = \log[N(\mathbf{y}|\mathbf{0}, \sigma^2 I + K_{nn})]. \qquad (2)$$

Although the above GP approach is elegant, it is intractable for large datasets since the computations require the inversion of a matrix of size $n \times n$ which scales as $O(n^3)$. Thus, we need to consider approximate or sparse methods in order to deal with large datasets. Advanced sparse methods use a small set of m function points as support or inducing variables. This yields a time complexity that scales as $O(nm^2)$. Important issues in these methods involve the selection of the inducing variables and the hyperparameters. For reviews of current approaches see chapter 8 in (Rasmussen and Williams, 2006) and (Quiñonero-Candela and Rasmussen, 2005).

Suppose we wish to use m inducing variables to construct our sparse GP method. The inducing variables are latent function values evaluated at some inputs X_m. X_m can be a subset of the training inputs or auxiliary pseudo-points (Snelson and Ghahramani, 2006). Learning X_m and the hyperparameters $(\boldsymbol{\theta}, \sigma^2)$ is the crucial problem we need to solve in order to obtain a sparse GP method. An approximation to the true log marginal likelihood in eq. (2) can allow us to infer these quantities. The current state-of-the-art approximate marginal likelihood is given in the sparse pseudo-inputs GP method (SPGP) proposed in (Snelson and Ghahramani, 2006). A related objective function used in (Seeger et al., 2003) corresponds to the projected process approximation (PP). These approximate log marginal likelihoods have the form

$$F = \log[N(\mathbf{y}|\mathbf{0}, \sigma^2 I + Q_{nn})], \qquad (3)$$

where Q_{nn} is an approximation to the true covariance K_{nn}. In PP, $Q_{nn} = K_{nm}K_{mm}^{-1}K_{mn}$, i.e. the exact covariance is replaced by the Nyström approximation. Here, K_{mm} is the $m \times m$ covariance matrix on the inducing inputs, K_{nm} is the $n \times m$ cross-covariance matrix between training and inducing points and $K_{mn} = K_{nm}^T$. In SPGP, $Q_{nn} = \text{diag}[K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}] + K_{nm}K_{mm}^{-1}K_{mn}$, i.e. the Nyström approximation is corrected to be exact in the diagonal. By contrasting eq. (2) with (3), it is clear that F is obtained by modifying the GP prior. This implies that the inducing inputs X_m play the role of extra kernel hyperparameters (similar to $\boldsymbol{\theta}$) that parametrize the covariance matrix Q_{nn}. However because the prior has changed, continuous optimization of F with respect to X_m does not reliably approximate the exact GP model. Further, since F is heavily parametrized with the extra hyperparameters X_m, overfitting can occur especially when we jointly optimize over $(X_m, \boldsymbol{\theta}, \sigma^2)$.

In the next section, we propose a formulation for sparse GP regression that follows a different philosophy. Rather than modifying the exact GP model, we minimize a distance between the exact posterior GP and a variational approximation. The inducing inputs X_m become now variational parameters which are rigorously selected so as the distance is minimized.

3 VARIATIONAL LEARNING

We wish to define a sparse method that directly approximates the posterior GP mean and covariance functions in eq. (1). This posterior GP can be also described by the predictive Gaussian $p(\mathbf{z}|\mathbf{y}) = \int p(\mathbf{z}|\mathbf{f})p(\mathbf{f}|\mathbf{y})d\mathbf{f}$, where $p(\mathbf{z}|\mathbf{f})$ denotes the conditional prior over any finite set of function points $\mathbf{z}$. Suppose that we wish to approximate the above Bayesian integral by using a small set of m auxiliary inducing variables $\mathbf{f}_m$ evaluated at the pseudo-inputs X_m, which are independent from the training inputs. $\mathbf{f}_m$ are just function points drawn from the same GP prior as the training function values $\mathbf{f}$. By using the augmented joint model $p(\mathbf{y}|\mathbf{f})p(\mathbf{z}, \mathbf{f}_m, \mathbf{f})$, we equivalently write $p(\mathbf{z}|\mathbf{y})$ as

$$p(\mathbf{z}|\mathbf{y}) = \int p(\mathbf{z}|\mathbf{f}_m, \mathbf{f})p(\mathbf{f}|\mathbf{f}_m, \mathbf{y})p(\mathbf{f}_m|\mathbf{y})d\mathbf{f}d\mathbf{f}_m. \quad (4)$$

true GP

Suppose now that $\mathbf{f}_m$ is a sufficient statistic for the parameter $\mathbf{f}$ in the sense that $\mathbf{z}$ and $\mathbf{f}$ are independent given $\mathbf{f}_m$, i.e. it holds $p(\mathbf{z}|\mathbf{f}_m, \mathbf{f}) = p(\mathbf{z}|\mathbf{f}_m)$. The above can be written as

$$\begin{aligned} q(\mathbf{z}) &= \int p(\mathbf{z}|\mathbf{f}_m)p(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m)d\mathbf{f}d\mathbf{f}_m \\ &= \int p(\mathbf{z}|\mathbf{f}_m)\phi(\mathbf{f}_m)d\mathbf{f}_m = \int q(\mathbf{z}, \mathbf{f}_m)d\mathbf{f}_m, \quad (5) \end{aligned}$$

where $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{y})$ and $\phi(\mathbf{f}_m) = p(\mathbf{f}_m|\mathbf{y})$. Here, $p(\mathbf{f}|\mathbf{f}_m) = p(\mathbf{f}|\mathbf{f}_m, \mathbf{y})$ is true since $\mathbf{y}$ is a noisy version of $\mathbf{f}$ and because of the assumption we made that any $\mathbf{z}$ is conditionally independent from $\mathbf{f}$ given $\mathbf{f}_m$[1]. In practise it is difficult to find inducing variables $\mathbf{f}_m$ that are sufficient statistics. Thus, we expect $q(\mathbf{z})$ to be only an approximation to $p(\mathbf{z}|\mathbf{y})$. In such case, we can choose $\phi(\mathbf{f}_m)$ to be a "free" variational Gaussian distribution, where in general $\phi(\mathbf{f}_m) \neq p(\mathbf{f}_m|\mathbf{y})$, that depends on a mean vector $\boldsymbol{\mu}$ and a covariance matrix A. By using eq. (5), we can write down the approximate posterior GP mean and covariance functions as follows

$$\begin{aligned} m^q_{\mathbf{y}}(\mathbf{x}) &= K_{\mathbf{x}m}K^{-1}_{mm}\boldsymbol{\mu}, \quad (6) \\ k^q_{\mathbf{y}}(\mathbf{x}, \mathbf{x}') &= k(\mathbf{x}, \mathbf{x}') - K_{\mathbf{x}m}K^{-1}_{mm}K_{m\mathbf{x}'} + K_{\mathbf{x}m}BK_{m\mathbf{x}'}, \end{aligned}$$

where $B = K^{-1}_{mm}AK^{-1}_{mm}$. The above defines the general form of the sparse posterior GP which is computed in $O(nm^2)$. The question that now arises is how do we select the ϕ distribution, i.e. ($\boldsymbol{\mu}$, A), and the inducing inputs X_m. Next we describe a variational method that allows to jointly specify these quantities and treat X_m as a variational parameter which is rigorously selected by minimizing the KL divergence.

A principled procedure to specify ϕ and the inducing inputs X_m is to form the variational distribution $q(\mathbf{f})$ and the exact posterior $p(\mathbf{f}|\mathbf{y})$ on the training function values $\mathbf{f}$, and then minimize a distance between these two distributions. Equivalently, we can minimize a distance between the augmented true posterior $p(\mathbf{f}, \mathbf{f}_m|\mathbf{y})$ and the augmented variational posterior $q(\mathbf{f}, \mathbf{f}_m)$ where clearly from eq. (5) $q(\mathbf{f}, \mathbf{f}_m) = p(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m)$. The augmented true posterior is associated with the augmented joint model

$$p(\mathbf{y}, \mathbf{f}, \mathbf{f}_m) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{f}_m)p(\mathbf{f}_m), \quad (7)$$

which is equivalent to the initial model $p(\mathbf{y}, \mathbf{f}) = p(\mathbf{f}|\mathbf{y})p(\mathbf{f})$, since by marginalizing out $\mathbf{f}_m$ from the former we always recover the latter. In particular, notice that the conditional prior $p(\mathbf{f}|\mathbf{f}_m)$ and the marginal prior $p(\mathbf{f}_m)$ depend on the specific values of the inducing inputs X_m. However, this dependence never affects the posterior $p(\mathbf{f}|\mathbf{y})$ or the marginal likelihood $p(\mathbf{y})$. Hence, the augmented representation has a set of "free" parameters X_m which can be treated as variational parameters as opposed to the model parameters.

To determine the variational quantities (X_m, ϕ), we minimize the KL divergence $\mathrm{KL}(q(\mathbf{f}, \mathbf{f}_m)||p(\mathbf{f}, \mathbf{f}_m|\mathbf{y}))$. This minimization is equivalently expressed as the maximization of the following variational lower bound of the true log marginal likelihood:

$$F_V(X_m, \phi) = \int p(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m) \log \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}_m)}{\phi(\mathbf{f}_m)} d\mathbf{f}d\mathbf{f}_m, \quad (8)$$

where the term $p(\mathbf{f}|\mathbf{f}_m)$ inside the log cancels out. We can firstly maximize the bound by analytically solving for the optimal choice of the variational distribution ϕ. The bound after this maximization is

$$F_V(X_m) = \log \left[N(\mathbf{y}|\mathbf{0}, \sigma^2 I + Q_{nn})\right] - \frac{1}{2\sigma^2}Tr(\widetilde{K}), \quad (9)$$

where $Q_{nn} = K_{nm}K^{-1}_{mm}K_{mn}$ and $\widetilde{K} = \mathrm{Cov}(\mathbf{f}|\mathbf{f}_m) = K_{nn} - K_{nm}K^{-1}_{mm}K_{mn}$. Details of the derivation of this bound are given in a technical report (Titsias, 2009). The novelty of the above objective function is that it contains a regularization trace term: $-\frac{1}{2\sigma^2}Tr(\widetilde{K})$. This clearly differentiates F_V from all marginal likelihoods, described by eq. (3), that were previously applied to sparse GP regression. We will analyze the trace term shortly.

[1] From $p(\mathbf{z}|\mathbf{f}_m, \mathbf{y}) = \frac{\int p(\mathbf{y}|\mathbf{f})p(\mathbf{z}, \mathbf{f}_m, \mathbf{f})d\mathbf{f}}{\int p(\mathbf{y}|\mathbf{f})p(\mathbf{z}, \mathbf{f}_m, \mathbf{f})d\mathbf{f}d\mathbf{z}}$ and by using the fact $p(\mathbf{z}|\mathbf{f}_m, \mathbf{f}) = p(\mathbf{z}|\mathbf{f}_m)$, the result follows.

The quantity in eq. (9) is computed in $O(nm^2)$ time and is a lower bound of the true log marginal likelihood for any value of the inducing inputs X_m. Further maximization of the bound can be achieved by optimizing over X_m and optionally over the number of these variables. Note that the inducing inputs determine the flexibility of the variational distribution $q(\mathbf{f}, \mathbf{f}_m) = p(\mathbf{f}|\mathbf{f}_m)\phi(\mathbf{f}_m)$ since by tuning X_m we adapt both $p(\mathbf{f}|\mathbf{f}_m)$ and the underlying optimal distribution ϕ^*. To compute this optimal ϕ^*, we differentiate eq. (8) with respect to $\phi(\mathbf{f}_m)$ without imposing any constraints. This gives:

$$\phi^*(\mathbf{f}_m) = N(\mathbf{f}_m|\boldsymbol{\mu}, A), \tag{10}$$

where $\boldsymbol{\mu} = \sigma^{-2}K_{mm}\Sigma K_{mn}\mathbf{y}$, $A = K_{mm}\Sigma K_{mm}$ and $\Sigma = (K_{mm} + \sigma^{-2}K_{mn}K_{nm})^{-1}$. This now fully specifies our variational GP and we can use eq. (6) to make predictions in unseen input points. Clearly, the predictive distribution is exactly the one used by the projected process (PP) that has been previously proposed in (Csato and Opper, 2002; Seeger et al., 2003). Thus, as far as the predictive distribution is concerned the above method is equivalent to PP.

However, the variational method is very different to PP and SPGP as far as the selection of the inducing inputs and the kernel hyperparameters is concerned. This is because of the extra regularization term that appears in the bound in eq. (9) and does not appear in the approximate log marginal likelihoods used in PP (Seeger et al., 2003) and SPGP (Snelson and Ghahramani, 2006). As discussed in section 2, for the latter objective functions, the role of X_m is to form a set of extra kernel hyperparameters. In contrast, for the lower bound, the inputs X_m become variational parameters due to the KL divergence that is minimized.

To look into the functional form of the bound, note that F_V is the sum of the PP log likelihood and the regularization trace term $-\frac{1}{2}\sigma^{-2}Tr(\widetilde{K})$. Thus, F_V attempts to maximize the PP log likelihood and simultaneously minimize the trace $Tr(\widetilde{K})$. $Tr(\widetilde{K})$ represents the total variance of the conditional prior $p(\mathbf{f}|\mathbf{f}_m)$ which also corresponds to the squared error of predicting the training latent values $\mathbf{f}$ from the inducing variables $\mathbf{f}_m$: $\int p(\mathbf{f}, \mathbf{f}_m)||K_{nm}K_{mm}^{-1}\mathbf{f}_m - \mathbf{f}||^2 d\mathbf{f}d\mathbf{f}_m$. When $Tr(\widetilde{K}) = 0$, the Nyström approximation is exact, i.e. $K_{nn} = K_{nm}K_{mm}^{-1}K_{mn}$, which means that the inducing variables become sufficient statistics and we can reproduce exactly the full GP prediction. Note that the trace $Tr(\widetilde{K})$ itself has been used as a criterion for selecting the inducing points from the training data in (Smola and Schölkopf, 2000) and is similar to the criterion used in (Lawrence et al., 2002).

When we maximize the variational lower bound, the hyperparameters $(\sigma^2, \boldsymbol{\theta})$ are regularized. It is easy to see how this is achieved for the noise variance σ^2. At a local maxima, σ^2 satisfies:

$$\sigma^2 = \frac{1}{n}\int_{\mathbf{f}_m} \phi^*(\mathbf{f}_m)||\mathbf{y} - \boldsymbol{\alpha}||^2 d\mathbf{f}_m + \frac{1}{n}Tr(\widetilde{K}), \tag{11}$$

where $||\mathbf{z}||$ denotes the Euclidean norm and $\boldsymbol{\alpha} = \mathbb{E}[\mathbf{f}|\mathbf{f}_m] = K_{nm}K_{mm}^{-1}\mathbf{f}_m$. This decomposition reveals that the obtained σ^2 will be equal to the estimated "actual" noise plus a "correction" term that is the average squared error of predicting the training latent values from the inducing variables.

So far we assumed that the inducing inputs are selected by applying gradient-based optimization. However, this can be difficult in high dimensional input spaces as the number of variables becomes very large. Further, the kernel function might not be differentiable with respect to the inputs. In such cases we can still apply the variational method by selecting the inducing inputs from the training inputs. An important property of this discrete optimization scheme is that F_V monotonically increases when we greedily select inducing inputs and adapt the hyperparameters. Next we discuss this greedy selection method.

3.1 GREEDY SELECTION

Let $m \subset \{1, \ldots, n\}$ be the indices of a subset of data that are used as the inducing variables. The training points that are not part of the inducing set are indexed by $n-m$ and are called the remaining points, e.g. $\mathbf{f}_{n-m}$ denotes the remaining latent function values. The variational method is applied similarly to the pseudo-inputs case. Assuming the variational distribution $q(\mathbf{f}) = p(\mathbf{f}_{n-m}|\mathbf{f}_m)\phi(\mathbf{f}_m)$, we can express a variational bound that has the same form as the bound in eq. (9) with the only difference that $\widetilde{K} = \mathrm{Cov}(\mathbf{f}_{n-m}|\mathbf{f}_m)$.

The selection of inducing variables among the training data requires a prohibitive combinatorial search. A suboptimal solution can be based on a greedy selection scheme where we start with an empty inducing set $m = \emptyset$ and a remaining set $n - m = \{1, \ldots, n\}$. At each iteration, we add a training point $j \in J \subset n-m$, where J is a randomly chosen working set, into the inducing set that maximizes the selection criterion Δ_j.

It is important to interleave the greedy selection process with the adaption of the hyperparameters $(\sigma^2, \boldsymbol{\theta})$. This can be viewed as an EM-like algorithm; at the E step we add one point into the inducing set and at the M step we update the hyperparameters. To obtain a reliable convergence, the approximate marginal likelihood must monotonically increase at each E or M step. The PP and SPGP log likelihoods do not satisfy such a requirement because they can also decrease as we add points into the inducing set. In contrast,

the bound F_V is guaranteed to monotonically increase since now the EM-like algorithm is a variational EM. To clarify this, we state the following proposition.

Proposition 1. Let $(X_m, \mathbf{f}_m)$ be the current set of inducing points and m the corresponding set of indices. Any point $i \in n - m$ added into the inducing set can never decrease the lower bound.

Proof: Before the new point $(f_i, \mathbf{x}_i)$ is added, the variational distribution is $p(\mathbf{f}_{n-m}|\mathbf{f}_m)\phi^*(\mathbf{f}_m) = p(\mathbf{f}_{n-(m\cup i)}|f_i, \mathbf{f}_m)p(f_i|\mathbf{f}_m)\phi^*(\mathbf{f}_m)$. When we add the new point, the term $p(f_i|\mathbf{f}_m)\phi^*(\mathbf{f}_m)$ is replaced by the optimal $\phi^*(f_i, \mathbf{f}_m)$ distribution. This can either increase the lower bound or leave it invariant. A more detailed proof is given in (Titsias, 2009).

A consequence of the above proposition is that the greedy selection process monotonically increases the lower bound and this holds for any possible criterion Δ. An obvious choice is to use F_V as the criterion, which can be evaluated in $O(nm)$ time for any candidate point in the working set J. Such a selection process maximizes the decrease in the divergence $\mathrm{KL}(q(\mathbf{f})||p(\mathbf{f}|\mathbf{y}))$.

4 COMPARISON

In this section we compare the lower bound F_V, the PP and the SPGP log likelihood in some toy problems. All these functions are continuous with respect to $(X_m, \sigma^2, \boldsymbol{\theta})$ and can be maximized using gradient-based optimization.

Our working example will be the one-dimensional dataset[2] considered in Snelson and Ghahramani (2006) that consists of 200 training points; see Figure 1. We train a sparse GP model using the squared exponential kernel defined by $\sigma_f^2 \exp(-\frac{1}{2\ell^2}||x_i - x_j||^2)$. Since the dataset is small and the full GP model is tractable, we compare the sparse approximations with the exact GP prediction. The plots in the first row of Figure 1 show the predictive distributions for the three methods assuming 15 inducing inputs. The left plot displays the mean prediction with two-standard error bars (shown as blue solid lines) obtained by the maximization of F_V. The prediction of the full GP model is displayed using dashed red lines. The middle plot shows the corresponding solution found by PP and the right plot the solution found by SPGP. The prediction obtained by the variational method almost exactly reproduces the full GP prediction. The final value of the variational lower bound was -55.5708, while the value of the maximized true log marginal likelihood was -55.5647. Further, the estimated hyperparameters found by F_V match the hyperparameters found by maximizing the true log marginal likelihood. In contrast, training the sparse model using the PP log likelihood gives a poor approximation. The SPGP method gave a much more satisfactory answer than PP although not as good as the variational method.

To consider a more challenging problem, we decrease the number of the original 200 training examples by maintaining only 20 of them[3]. We repeat the experiment above using exactly the same setup. The second row of Figure 1, displays the predictive distributions of the three methods. The prediction of the variational method is identical to the full GP prediction and the hyperparameters match those obtained by full GP training. On the other hand, the PP log likelihood leads to a significant overfitting of the training data since the mean curve interpolates the training points and the error bars are very noisy. SPGP provides a solution that significantly disagrees with the full GP prediction both in terms of the mean prediction and the errors bars. Notice that the width of the error bars found by SPGP varies a lot in different input regions. This nonstationarity is achieved by setting σ^2 very close to zero and modelling the actual noise by the heteroscedastic diagonal matrix $\mathrm{diag}[K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}]$. The fact that this diagonal matrix (the sum of its elements is the trace $Tr(\widetilde{K})$) is large indicates that the full GP model is not well approximated.

The reason PP and SPGP do not recover the full GP model when we optimize over $(X_m, \sigma^2, \boldsymbol{\theta})$ is not the local maxima. To clarify this point, we repeated the experiments by initializing the PP and SPGP log likelihoods to optimal inducing inputs and hyperparameters values where the later are obtained by full GP training. The predictions found are similar to those shown in Figure 1. A way to ensure that the full GP model will be recovered as we increase the number of inducing inputs is to select them from the training inputs. This, however, turns the continuous optimization problem into a discrete one and moreover PP and SPGP face the non-smooth convergence problem.

Regarding F_V, it is clear from section 3 that by maximizing over X_m we approach the full GP model in the sense of $\mathrm{KL}(q(\mathbf{f}, \mathbf{f}_m)|p(\mathbf{f}, \mathbf{f}_m|\mathbf{y}))$. Something less clear is that F_V efficiently regularizes the hyperparameters $(\sigma^2, \boldsymbol{\theta})$ so as overfitting is avoided. This is achieved by the regularization trace term: $-\frac{1}{2}\sigma^{-2}Tr(\widetilde{K})$. When $Tr(\widetilde{K})$ is large because there are not sufficiently many inducing variables, this term favours kernel parameters that give a smoother function. Also, when $Tr(\widetilde{K})$ is large the decomposition in eq. (11) implies that σ^2

[2] obtained from `www.gatsby.ucl.ac.uk/~snelson/`.

[3] The points were chosen from the original set according to the MATLAB command: X = X(1:10:end).

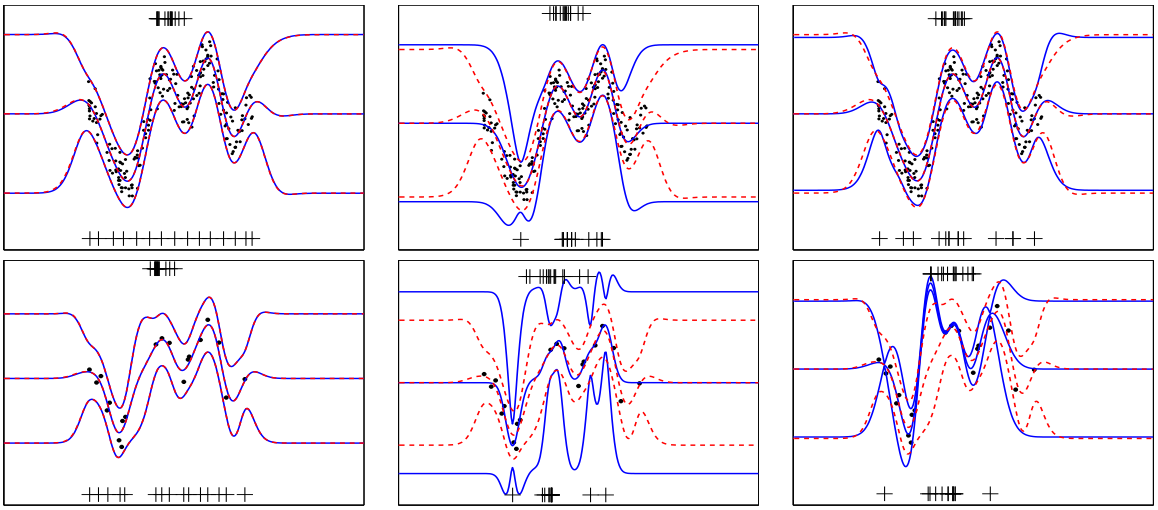

Figure 1: The first row corresponds to 200 training points and the second row to 20 training points. The first column shows the prediction (blue solid lines) obtained by maximizing F_V over the 15 pseudo-inputs and the hyperparameters. The full GP prediction is shown with red dashed lines. Initial locations of the pseudo-inputs are shown on the top as crosses, while final positions are given on the bottom as crosses. The second column shows the predictive distributions found by PP and similarly the third column for SPGP.

must increase as well. These properties are useful for avoiding overfitting and also imply that the prediction obtained by F_V will tend to be smoother than the prediction of the full GP model. In contrast, the PP and SPGP log likelihoods can find more flexible solutions than the full GP prediction which indicates that they are prone to overfitting.

5 EXPERIMENTS

In this section we compare the variational lower bound (VAR), the projected process approximate log likelihood (PP) and the sparse pseudo-inputs GP (SPGP) log likelihood in four real datasets. As a baseline technique, we use the subset of data (SD) method. For all sparse GP methods we jointly maximize the alternative objective functions w.r.t. hyperparameters $(\boldsymbol{\theta}, \sigma^2)$ and the inducing inputs X_m using the conjugate gradients algorithm. X_m is initialized to a randomly chosen subset of training inputs. In each run all methods are initialized to the same inducing inputs and hyperparameters. The performance criteria we use are the standardized mean squared error (SMSE), given by $\frac{1}{T}\frac{||\mathbf{y}_* - \mathbf{f}_*||^2}{var(\mathbf{y}_*)}$, and the standardized negative log probability density (SNLP) as defined in (Rasmussen and Williams, 2006). Smaller values for both error measures imply better performance. In all the experiments we use the squared-exponential kernel with varied length-scale.

Firstly, we consider the Boston-housing dataset, which consists of 455 training examples and 51 test examples. Since the dataset is small, full GP training is tractable. In the first experiment, we fix the parameters $(\boldsymbol{\theta}, \sigma^2)$ to values obtained by training the full GP model. Thus we can investigate the difference of the methods solely on how the inducing inputs are selected. We rigorously compare the methods by calculating the moments-matching divergence $\mathrm{KL}(p(\mathbf{f}_*|\mathbf{y})||q(\mathbf{f}_*))$ between the true test posterior $p(\mathbf{f}_*|\mathbf{y})$ and each of the approximate test posteriors. For the SPGP method the approximate test posterior distribution is computed by using the exact test conditional $p(\mathbf{f}_*|\mathbf{f}_m)$. Figure 2(a) show the KL divergence as the number of inducing points increases. Means and one-standard error bars were obtained by repeating the experiment 10 times. Note that only the VAR method is able to match the full GP model; for around 200 points we closely match the full GP prediction. Interestingly, when the inducing inputs are initialized to all training inputs, i.e. $X_m = X$, PP and SPGP still give a different solution from the full GP model despite the fact that the hyperparameters are kept fixed to the values of the full GP model. The reason this is happening is that they are not lower bounds to the true log marginal likelihood and as shown in Figure 2(c) they become upper bounds. To show that the effective selection of the inducing inputs achieved by VAR is not a coincidence, we compare it with the case where the inputs are kept fixed to their initial randomly selected training inputs. Figure 2(b) displays the evolution of the KL divergence for the VAR, the random selection plus PP (RSPP) and the SD method. Note that the only difference between VAR and RSPP is that VAR

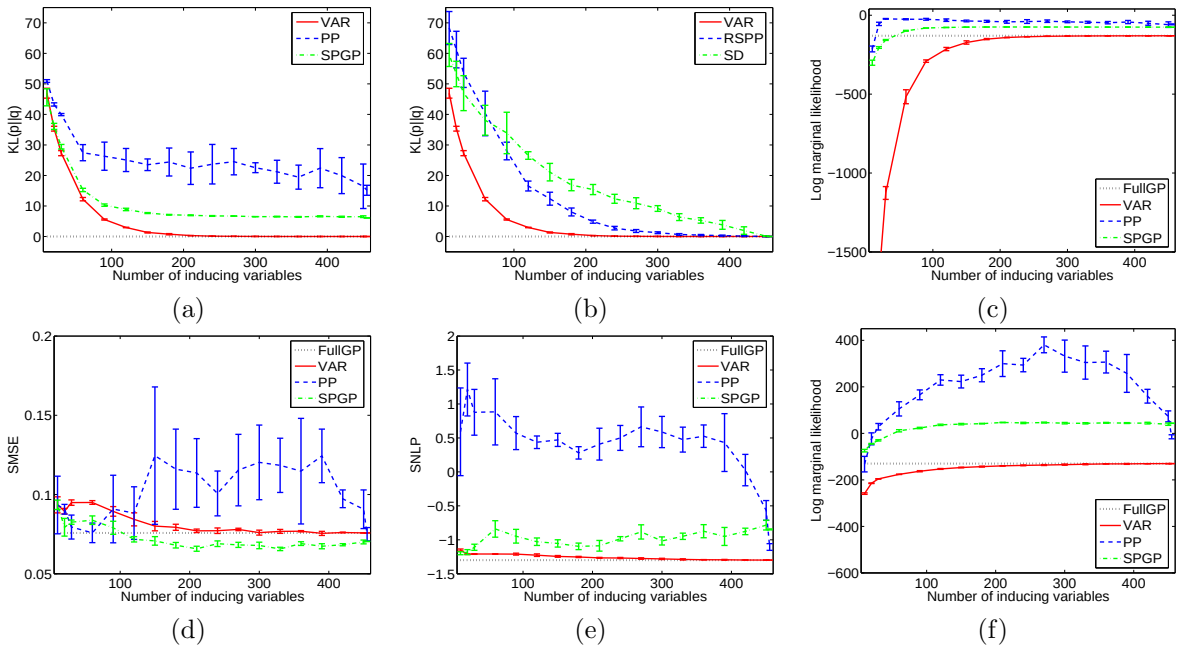

Figure 2: (a) show the KL divergence as the number of inducing variables increases for the VAR the PP and SPGP methods. Similarly (b) show the divergence for the VAR, RSPP and SD methods. (c) displays the approximate log marginal likelihoods; the true log marginal likelihood value is displayed by using the dotted horizontal line. (d) and (e) show the SMSE and SNLP errors (obtained by joint learning hyperparameters and inducing-inputs) against the number of inducing variables. (f) shows the corresponding log marginal likelihoods.

optimizes the lower bound over the initial values of the inducing inputs, while RSPP just keep them fixed. Clearly RSPP significantly improves over the SD prediction, and VAR significantly improves over RSPP.

In a second experiment, we jointly learn inducing variables and hyperparameters and compare the methods in terms of the SMSE and SNLP errors. The results are displayed in the second row of Figure 2. Note that the PP and SPGP methods achieve a much higher log likelihood value (Figure 2(f)) than the true log marginal likelihood. However, the error measures clearly indicate that the PP log likelihood significantly overfits the data. SPGP gives better SMSE error than the full GP model but it overfits w.r.t. the SNLP error. The variational method matches the full GP model.

We now consider three large datasets: the KIN40K dataset, the SARCOS and the ABALONE datasets[4] that have been widely used before. Note that the ABALONE dataset is small enough so as we will be able to train the full GP model. The inputs were normalized to have zero mean and unit variance on the training set and the outputs were centered so as to have zero mean on the training set. For the KIN40K and the SARCOS datasets, the SD method was obtained in a subset of 2000 training points. We vary the size of the inducing variables in powers of two from 16 to 1024. For the SARCOS dataset, the experiment for 1024 was not performed since is was unrealistically expensive. All the objective functions were jointly maximized over inducing inputs and hyperparameters. The experiment was repeated 5 times. Figure 3 shows the results.

From the plots in Figure 3, we can conclude the following. The PP log likelihood is significantly prone to overfitting as the SNLP errors clearly indicate. However, note that in the KIN40K and SARCOS datasets, PP gave the best performance w.r.t. to SMSE error. This is probably because of the ability of PP to interpolate the training examples that can lead to good SMSE error when the actual observation noise is low. SPGP often has the worst performance in terms of the SMSE error and almost always the best performance in terms of the SNLP error. In the ABALONE dataset, SPGP had significantly better SNLP error than the full GP model. Since the SNLP error depends on the predictive variances, we believe that the good performance of SPGP is due to its heteroscedastic ability. For example, in the KIN40K dataset, SPGP makes σ^2 almost zero and thus the actual noise in the likelihood is modelled by the heteroscedastic covariance

[4] KIN40K: 10000 training, 30000 test, 8 attributes, ida.first.fraunhofer.de/ anton/data.html.
SARCOS: 44,484 training, 4,449 test, 21 attributes, www.gaussianprocess.org/gpml/data/.
ABALONE: 3,133 training, 1,044 test, 8 attributes, www.liaad.up.pt/ ltorgo/Regression/DataSets.html.

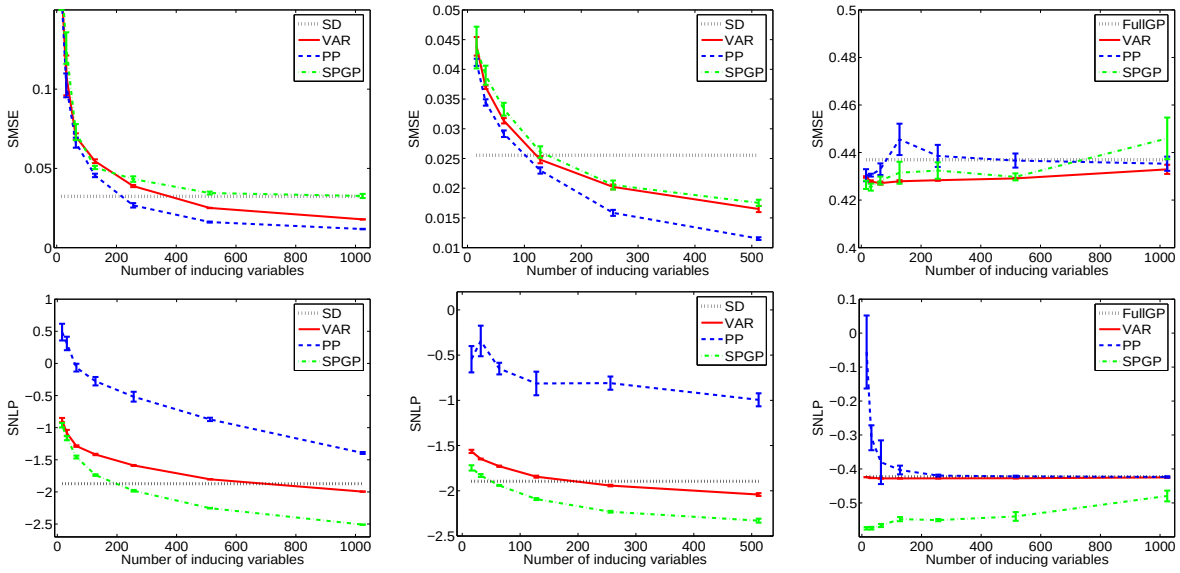

Figure 3: The first column displays the SMSE (top) and SNLP (bottom) errors for the KIN40K dataset with respect to the number of inducing points. The second column shows the corresponding plots for the SARCOS dataset and similarly the third column shows the results for the ABALONE dataset.

diag$[K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}]$. The fact that the latter term is large may indicate that the full GP model is not well approximated. Finally the variational method has good performance. VAR never had the worst performance and it didn't exhibit overfitting. The examples in section 4, the Boston-housing and the ABALONE dataset indicate that the VAR method remains much closer to the full GP model than the other methods.

6 CONCLUSION

We proposed a variational framework for sparse GP regression that can reliably learn inducing inputs and hyperparameters by minimizing the KL divergence between the true posterior GP and an approximate one. This method can be more generally applicable. Currently we apply this technique to classification. An interesting topic for the future is to apply this method to GP models that assume multiple latent functions.

Acknowledgments

I am grateful to Neil Lawrence for his help. This work is funded by EPSRC Grant No EP/F005687/1 "Gaussian Processes for Systems Identification with Applications in Systems Biology".

References

Csato, L. and Opper, M. (2002). Sparse online Gaussian processes. *Neural Computation*, 14:641–668.

Lawrence, N. D., Seeger, M., and Herbrich, R. (2002). Fast sparse Gaussian process methods: the informative vector machine. In *Neural Information Processing Systems, 13*. MIT Press.

Quiñonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Schwaighofer, A. and Tresp, V. (2003). Transductive and inductive methods for approximate Gaussian process regression. In *Neural Information Processing Systems 15*. MIT Press.

Seeger, M. (2003). *Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations*. PhD thesis, University of Edinburgh.

Seeger, M., Williams, C. K. I., and Lawrence, N. D. (2003). Fast forward selection to speed up sparse Gaussian process regression. In *Ninth International Workshop on Artificial Intelligence*. MIT Press.

Smola, A. J. and Bartlett, P. (2001). Sparse greedy Gaussian process regression. In *Neural Information Processing Systems, 13*. MIT Press.

Smola, A. J. and Schölkopf, B. (2000). Sparse greedy matrix approximations for machine learning. In *International Conference on Machine Learning*.

Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian process using pseudo-inputs. In *Neural Information Processing Systems, 13*. MIT Press.

Titsias, M. K. (2009). Variational Model Selection for Sparse Gaussian Process Regression. Technical report, School of Computer Science, University of Manchester.

Williams, C. K. I. and Seeger, M. (2001). Using the Nyström method to speed up kernel machines. In *Neural Information Processing Systems 13*. MIT Press.