Supplementary material for the Efficient Sampling for Gaussian Process Inference using Control Variables

Abstract

This extra material provides a description of the local region sampling algorithm for GP models.

1 Introduction

The MCMC algorithm we consider is the general Metropolis-Hastings (MH) algorithm. Suppose we wish to sample from the posterior GP process. The MH algorithm forms a Markov chain. We initialize $\mathbf{f}^{(0)}$ and we consider a proposal distribution $Q(\mathbf{f}^{(t+1)}|\mathbf{f}^{(t)})$ that allows us to draw a new state given the current state. The new state is accepted with probability $\min(1, A)$ where

$$A = \frac{p(\mathbf{y}|\mathbf{f}^{(t+1)})p(\mathbf{f}^{(t+1)})}{p(\mathbf{y}|\mathbf{f}^{(t)})p(\mathbf{f}^{(t)})} \frac{Q(\mathbf{f}^{(t)}|\mathbf{f}^{(t+1)})}{Q(\mathbf{f}^{(t+1)}|\mathbf{f}^{(t)})}. \tag{1}$$

To apply this generic algorithm, we need to choose the proposal distribution Q. For GP models, finding a good proposal distribution is challenging since $\mathbf{f}$ is high dimensional and the posterior distribution can be highly correlated.

Two extreme options for the proposal distribution Q is to sample from the GP prior or apply Gibbs sampling where we iteratively draw samples from each posterior conditional density $p(f_i|\mathbf{f}_{-i}, \mathbf{y})$ with $\mathbf{f}_{-i} = \mathbf{f} \setminus f_i$. A similar algorithm to Gibbs sampling can be expressed by using as a proposal distribution the sequence of the conditional densities $p(f_i|\mathbf{f}_{-i})$[1]. We call this algorithm the Gibbs-like algorithm. Next we describe a modification of the Gibbs-like algorithm that is more efficient.

2 Sampling using local regions

To overcome the limitations of the Gibbs and Gibbs-like algorithm we can divide the domain of the function into regions and sample the entire function within each region. Assuming that the number of the regions depends mainly on the shape of the function and not on the discretization, this scheme can be more efficient.

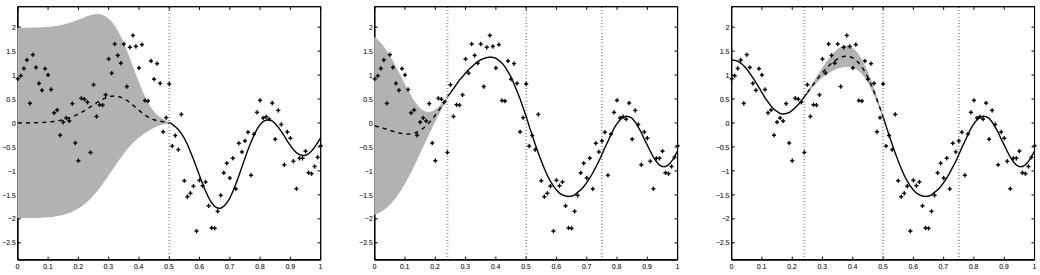
Let $\mathbf{f}_k$ denote the function values that belong to the local region k, where $k = 1, \ldots, M$ and $\mathbf{f}_1 \cup \ldots \cup \mathbf{f}_M = \mathbf{f}$. New values for the region k are proposed by drawing from the conditional GP prior $p(\mathbf{f}_k^{t+1}|\mathbf{f}_{-k}^{(t)})$, where $\mathbf{f}_{-k} = \mathbf{f} \setminus \mathbf{f}_k$, by conditioning on the remaining function values. $\mathbf{f}_k^{(t+1)}$ is accepted with probability $\min(1, A)$ where

$$A = \frac{p(\mathbf{y}|\mathbf{f}_k^{(t+1)}, \mathbf{f}_{-k}^{(t)})}{p(\mathbf{y}|\mathbf{f}_k^{(t)}, \mathbf{f}_{-k}^{(t)})}. \tag{2}$$

Sampling $\mathbf{f}_k$ is iterated between all different regions $k = 1, \ldots, M$. Note that the terms associated with the GP prior cancel out from the acceptance probability since sampling from the conditional prior ensures that any proposed sample is consistent with the prior smoothness requirement. Sampling from the GP prior and the Gibbs-like algorithm are special cases of the above algorithm.

To apply the above algorithm, we need to partition the function values $\mathbf{f}$ into clusters. This process of adapting the proposal distribution can be carried out during the burn-in sampling phase. If we start

[1] Thus we replace the proposal distribution $p(f_i|\mathbf{f}_{-i}, \mathbf{y})$ with the prior conditional $p(f_i|\mathbf{f}_{-i})$.



(a) (b) (c)

Figure 1: Illustration of the hierarchical clustering process. The panel in (a) shows the variance (displayed with shaded two standard errors bars) of the initial conditional GP prior where we condition on the right side of the function. Since the variance is high the generated local parts of the function will not fit the data often. Dividing the local input region in (a) into two smaller groups (plots (b) and (c)) results a decrease of the variance of the newly formed GP conditional priors and an increase of the acceptance rate.

with a small number of clusters, so as the acceptance rate is very low, our objective is to refine these initial clusters in order to increase the acceptance rate. Following the widely used heuristics [1] according to which desirable acceptance rates of MH algorithms are around 1/4, we require the algorithm to sample with acceptance rate larger than 1/4.

We obtain a initial partitioning of the vector $\mathbf{f}$ by clustering the inputs X using the kmeans algorithm. Then we start the simulation and we observe the local acceptance rate r_k associated with the proposal $p(\mathbf{f}_k|\mathbf{f}_{-k})$. Each r_k provides information about the variance of the proposal distribution relative to the local characteristics of the function. A small r_k implies that $p(\mathbf{f}_k|\mathbf{f}_{-k})$ has high variance and most of the generated samples are outside of the support of the GP posterior process; see Figure 1 for an illustrative example. Thus, when r_k is small, we split the cluster k into two clusters by locally applying the kmeans algorithm using all the inputs previously assigned to the initial cluster k. Clusters that have high acceptance rate are unchanged. This hierarchical partitioning process is recursively repeated until all the current clusters exhibit a local acceptance rate larger than a predefined threshold (this was set to 1/4 for all our experiments). The above partitioning process is supervised since the information provided by the MH steps is used to decide which clusters need to be split into smaller clusters.

Once the adaption of the proposal distribution is ended, we can start sampling from the posterior GP process. The final form of the proposal distribution is a partition of the vector $\mathbf{f}$ into M disjoint groups and the conditional GP prior is the proposal distribution for each group.

References

[1] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis*. Chapman and Hall, 2004.