Scalability and uncertainty of Gaussian processes

Hadji, M.A.

CHAPTER 5
Simulation study

Abstract. In this chapter we investigate the numerical properties of the different Gaussian process regression techniques using distributed methods. Distributed methods use a divide-and-conquer strategy: the supposedly large data set is divided among m machines. This strategy helps reducing the computational costs of the typical Bayesian non-parametric regression. It should be noted that there exist multiple ways of partitioning the data among the machines

§5.1 Distributed GP regression

Gaussian process regression is arguably a very useful tool in machine learning since it can elegantly capture complex relationships in data (Rasmussen and Williams, 2006). However, it scales very poorly in computation and memory ($O(n^3)$ and $O(n^2)$ respectively, where n is the number of data points). This limitation inspired different approximation approaches, among which the *divide-and-conquer* strategy where the design is partitioned into m "expert" machines; then, the kth partition, with $k \in \{1, ..., m\}$ of size n_k is modeled by the "expert" to which it was assigned. Different models arose according to the way the data is allotted to the expert machines. A uniformly random partition model (see (Cao and Fleet, 2014) and (Tresp, 2000)) are built by allotting each machine a random subset of the data of size n/m in order to independently compute predictive distributions which will be aggregated. These have been shown to not only be Kolmogorov inconsistent (Samo and Roberts, 2016), but also to have posterior which contracts at a sub-optimal rates (Szabó and van Zanten, 2019).

On the other hand, spatial partition models are based on a division of the design space into non-overlapping region. Each machine is assigned a specific region and inference is made using the data in this region. For instance, the Naive-Local-Experts model (Kim et al., 2005) models each region with an independent GP. Its main drawback is the introduction of discontinuities in the prediction at the border of each region. There exist multiple ways to address the issue. Patched GPs (see (Park and Huang, 2016) and (Park and Apley, 2018)) for instance impose continuity constraints such that two adjacent local GPs are patched to share the nearly identical predictions on the boundary. Two-step Mixtures introduce a latent variable to the model which dynamically selects an expert to draw prediction on a given point (Tresp, 2001), (Rasmussen and Ghahramani, 2002), (Meeds and Osindero, 2006). Recently, hierarchical spatial partitioning models (Ng and Deisenroth, 2014) have been developed. They

typically result in posterior predictive distributions in the form of an average of all the "expert" predictions with a weight supposed to indicate how the confidence level of each prediction.

§5.1.1 Uniformly random

The data can be randomly split among the m machines. Each machine will receive a random sub-sample of the data, and will therefore solve a smaller version of the initial regression by computing a local posterior. The different posteriors will then be aggregated to form a global posterior. As seen in (Szabó and van Zanten, 2019), some adjustments should be made locally in order to obtain theoretical guaranties for this method. In this simulation study, we chose to adjust the local prior by raising it to the power $1/m$ and to average the local posteriors. However, (Szabó and van Zanten, 2019) also shows that despite the modifications on the local prior, adaptation leads to sub-optimal contraction rates and bad coverage for some true functions.

§5.1.2 Spatial

The data can also be split into subsets of the design point set. Each machine will receive data such that the design points belong to a certain sub-region. The sub-regions are not to overlap. The machine can then be seen as local experts; each expert is specialized in one particular sub-region of design points. A draw from the global posterior will thus consist of the local posterior draws restricted to their corresponding intervals and pasted together. Due to the localized structure, there is no need for alterations in the local prior. Moreover, this structure allows the local posterior to adapt to the unknown smoothness as we showed in Chapter 4. Unfortunately, the global posterior obtained by this procedure contains unwanted discontinuities at the border of each regions.

§5.1.3 Weighted-average model

One can note that both global posteriors produce samples in the form of a weighted average of the local samples. In the first scenario, a global posterior draw θ is defined as

$$\theta(x) = \frac{1}{m} \sum_{k=1}^{m} \theta^{(k)}(x),$$

for all $x \in \mathcal{X}$ where the $\theta^{(k)}$'s are local draws. In the second scenario, a global posterior draw can be written as

$$\theta(x) = \sum_{k=1}^{m} 1_{\mathcal{D}_k}(x) \theta^{(k)}(x),$$

where $\mathcal{D}_k$'s are the sub-regions into which the design points are partitioned. This observation explains the discontinuities in the latter case, since the weights are discontinuous themselves. In order to palliate this problem, we propose using data-driven weight functions which are both continuous and close to indicator functions. One can

find in the literature (Ng and Deisenroth, 2014) weights proportional to the inverse variances:

$$\theta(x) = \Sigma(x) \sum_{k=1}^{m} \frac{\theta^{(k)}(x)}{\sigma_k^2(x)},$$

where the σ_j^2's are the local posterior variance and $\Sigma(x) = \left(\sum_{k=1}^{m} \sigma_k^{-2}(x)\right)^{-1}$. Although these weights are data-driven and continuous, we will see that the corresponding global posterior exhibit sub-optimal asymptotic characteristics in an adaptive setting. Indeed, adapting to local smoothnesses may lead to shrinking variances for some machines, which in turn will lead the corresponding weight to be overly large even outside of the expert's domain. That is to say that experts are overly confident about the behavior of the true function in the whole space when this function is particularly smooth in this expert's domain. That is why we propose a modification of the weight functions so that they shrink quickly outside of their corresponding region minimizing the behavior of indicator functions. Namely, we choose

$$\theta(x) = W(x) \sum_{k=1}^{m} w_k(x)\theta^{(k)}(x),$$

where $w_k(x) = e^{-m^2(x-c_k)^2}/\sigma_k^2(x)$ with c_k being the center of gravity of $\mathcal{D}_k$, and $W(x) = \left(\sum_{k=1}^{m} w_k(x)\right)^{-1}$. These weights are both continuous and data-driven.

§5.2 Numerical study

§5.2.1 Simulated Data

First, we consider Gaussian process regression with simulated data that will allow us to compare the different distributed techniques to the non-distributed Gaussian regression, which will act as a benchmark. We see that if the smoothness of the true function is known and the Gaussian process parameters are chosen accordingly, all proposed distributed methods behave similarly. They present comparable L_2 distance between the true function and the posterior mean, and they portray similar coverage for their point-wise credible sets. Moreover, if the number of machines m does not grow too fast, these distributed methods are also similar to our benchmark. On the other hand, we show that in the adaptive setting, the way the data is distributed among the machines affect greatly the performance of the regression.

In this model we assume to observe n independent pairs of random variables $(X_1, Y_1), ...(X_n, Y_n)$, where

$$Y_i = \theta_0(X_i) + \varepsilon_i, \qquad \varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2), \quad X_i \overset{iid}{\sim} U(0, 1),$$

and the aim is to estimate the unknown non-parametric regression function θ_0. In the Bayesian approach we endow θ_0 with a Gaussian process prior with squared exponential kernel and estimate the tuning parameter using the MMLE. In addition, we wish to reduce the computational time by distributing the n pairs of random variables among m machines such that each machine only deals with the independent

pairs $(X_1^{(k)}, Y_1^{(k)}), ...(X_{n_k}^{(k)}, Y_{n_k}^{(k)})$ which represent a subset of the original sample and $n_k = n/m$ is the number of random variable pairs in the corresponding machine. All the posterior means, credible sets, and the empirical Bayes posteriors in the adaptive setting are all computed using the MatLab package gpml. Let us consider the function $\theta_0 \in L_2[0,1]$ given by the coefficients $\theta_{0,i} = i^{-3/2}\sin(i)$, for $i \geq 3$ and $\theta_{0,i} = 0$ otherwise, relative to the cosine eigenbasis $\psi_i(t) = \sqrt{2}\cos(\pi(i-1/2)t)$. Note that although the function lies outside of the self-similar function class, it has essentially the same behavior.

We take $\sigma^2 = 1$, but in the procedure it is considered to be unknown and estimated with the MMLE $\hat{\sigma}^2$. We plot in figures the true function (black), the posterior mean (colored), and the posterior point-wise credible intervals (shaded area) $[\hat{\theta}(x) - q_{0.025}\sqrt{\hat{c}(x,x)}, \hat{\theta}(x) + q_{0.025}\sqrt{\hat{c}(x,x)}]$, where $\hat{\theta}$ is the posterior mean, q_α the α-th quantile of the standard normal distribution and $\hat{c}(.,.)$ the posterior covariance kernel. We consider the non-distributive method (at the top) along with the four distributed methods proposed. We will compare the methods in different figures depending on the setting (non-adaptive or adaptive), the sample size ($n = 100, 500, 1000$ or 2000) and the number of machines ($m = 10$, $m \approx n^{1/3}$ or $m = n/100$).

We also investigate empirically the rate at which the posterior mean concentrates around the truth and the frequentist coverage probabilities of the point-wise credible sets by repeating the experiment 100 times and reporting the average integrated mean squared error and the frequency that the function at given points (we consider $x = (0.5, 0.41148, 0.31143)$ with $0.41148 = \text{argmin}_{x\in[0,1]}\theta_0(x)$ and $0.31143 = \text{argmax}_{x\in[0,1]}\theta_0(x)$) is included in the credible interval. See Tables 5.1afor the average L_2-norm between the posterior mean and the true function, and see Tables 5.2for the frequentist coverage of the credible sets.

The different methods we study are summed up in this table:

Method	*Description*
1	uniformly random partitioning + adjusting the prior with power $1/m$
2	spatial partitioning
3	spatial partitioning + inverse local post variance weights
4	spatial partitioning + inverse centered squared-exponential weights

Figure 5.1 illustrates that when m increases at sub-linear rate with n, the global posterior means obtained via the different methods are similar in a non-adaptive setting. Besides, these global posteriors means look similar to the traditional posterior mean. Nonetheless, the global posteriors as wholes do not behave similarly. **Method 2**, for instance, produces visible discontinuous predictions on the boundaries of subregions One can also see in Table 5.4 that global posteriors of distributed Bayesian regression take substantially less time to compute. It should also be taken into account that the all computations have been done sequentially; the parameters of all local posteriors have been computed and stored in the same computer. This may explain why adding more experts does not necessarily decreases the computation time. In practice, one can imagine that these running times might be reduced using multiple machines or cores, and that the effective time of the operation would roughly equal the present computation time divided by the number of machines.

Observe in Table 5.1a that the posterior mean in all distributed methods concen-

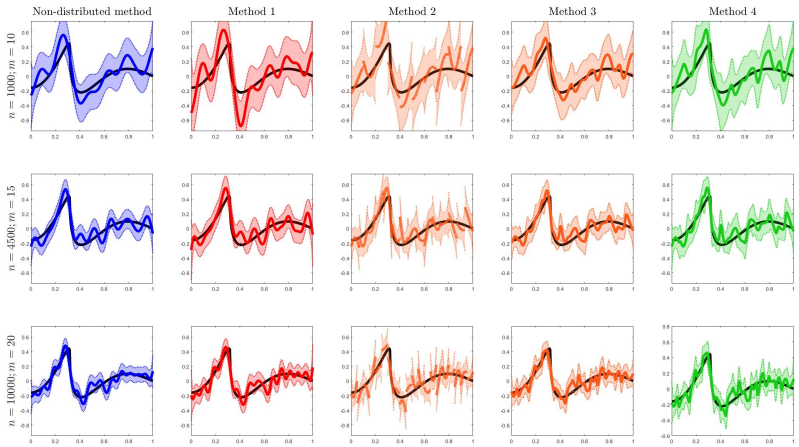

Figure 5.1: Non-adaptive posterior density for the function θ_0 (drawn in black) on $x \in [0,1]$. The posterior means are drawn by solid line, while the 95% point-wise credible sets are shaded between two dotted lines. In the first column, we plot the non-distributed method, in the second column the distributed method with random partitioning, the third column the distributed method with spatial partitioning, while in the fourth and fifth column we plot the distributed method with spatial partitioning with inverse variance weights and exponential weights respectively. From top to bottom the sample size is $n = 1000, 5000, 10000$ and the number of experts is $m = 10, 15, 20$.

	$n =$	1000	4500	10000	1000	4500	10000
	BM	**39.0041**	**23.5635**	**17.0517**	**49.9397**	**28.4795**	**20.3814**
$m = 10$	M 1	48.0590	26.1719	18.6091	64.2480	30.9439	22.2252
	M 2	47.5648	27.4998	19.5424	67.2354	31.2043	23.6621
	M 3	36.5442	24.0340	17.8270	56.6046	28.7473	22.2875
	M 4	42.1584	25.5088	18.3610	55.1067	29.5429	22.5838
$m \approx n^{1/3}$	M 1	48.0590	27.0015	19.2581	64.2480	31.9520	23.1445
	M 2	47.5648	29.4000	22.1400	67.2354	33.4671	26.8086
	M 3	36.5442	24.3732	18.6893	56.6046	29.5770	23.6461
	M 4	42.1584	26.7001	20.0192	55.1067	30.9875	24.7048
$m = \frac{n}{100}$	M 1	48.0590	36.5080	35.7912	64.2480	61.4936	84.4595
	M 2	47.5648	37.7861	32.7527	67.2354	43.1683	39.6620
	M 3	36.5442	24.9314	23.6126	56.6046	30.5618	23.9152
	M 4	42.1584	31.7063	25.9903	55.1067	36.9440	32.3062

(a) Average L_2 distance between θ_0 and the posterior mean for the squared exponential Gaussian process prior in a non-adaptive setting.

(b) Average L_2 credible ball radius for the squared exponential Gaussian process prior in a non-adaptive setting.

Table 5.1: BM: Benchmark, Non-distributed method. M 1: Random partitioning, M 2: Spatial partitioning, M 3: Spatial partitioning with inverse variance weights, M 4: Spatial partitioning with exponential weights. From left to right the sample size is $n = 1000, 4500, 10000$.

trates around the true function at a similar rate in a non-adaptive setting. When $m \lesssim \sqrt[3]{n}$, the contraction rate of the distributed posterior mean is virtually the same as the non-distributed case. However, as soon as the number of machine increases

		$x = 0.5$			$x = 0.41148$			$x = 0.31143$		
	$N =$	1000	4500	10000	1000	4500	10000	1000	4500	10000
$m = 10$	Benchmark	**0.99**	**0.96**	**0.95**	**0.96**	**0.91**	**1.00**	**0.57**	**0.48**	**0.30**
	Method 1	0.98	0.97	0.94	0.93	0.92	0.99	0.73	0.62	0.43
	Method 2	0.97	0.99	0.98	0.97	0.95	1.00	0.91	0.80	0.46
	Method 3	0.99	0.98	0.97	0.99	0.95	1.00	0.69	0.60	0.37
	Method 4	1.00	0.98	0.99	1.00	0.97	1.00	0.94	0.76	0.55
$m \approx \sqrt[3]{n}$	Method 1	0.98	0.95	0.97	0.93	0.96	0.98	0.73	0.66	0.49
	Method 2	0.97	0.97	0.98	0.97	0.95	0.98	0.91	0.67	0.45
	Method 3	0.99	0.99	0.97	0.99	0.95	0.99	0.69	0.47	0.28
	Method 4	1.00	0.97	0.99	1.00	0.97	0.98	0.94	0.63	0.52
$m = \frac{n}{100}$	Method 1	0.98	0.95	1.00	0.93	0.91	1.00	0.73	0.78	0.70
	Method 2	0.97	0.95	1.00	0.97	0.94	0.98	0.91	0.98	0.98
	Method 3	0.99	0.98	0.96	0.99	0.95	0.78	0.69	0.38	0.04
	Method 4	1.00	0.97	1.00	1.00	0.97	1.00	0.94	0.97	0.96

Table 5.2: Frequencies that $\theta_0(x)$ is inside of the corresponding credible interval for the squared exponential Gaussian process prior in a non-adaptive setting at given points $x \in \{0.5, 0.41148, 0.31143\}$. Benchmark: Non-distributed method. Method 1: Random partitioning, Method 2: Spatial partitioning, Method 3: Spatial partitioning with inverse variance weights, Method 4: Spatial partitioning with exponential weights. From left to right the sample size is $n = 1000, 4500, 10000$.

	$N =$	1000	4500	10000
$m = 10$	Benchmark	**0.93**	**0.96**	**0.95**
	Method 1	0.96	0.95	0.92
	Method 2	0.95	0.95	0.95
	Method 3	0.99	0.98	0.99
	Method 4	0.99	0.96	0.98
$m \approx \sqrt[3]{n}$	Method 1	0.96	0.93	0.93
	Method 2	0.95	0.98	0.99
	Method 3	0.99	0.99	1.00
	Method 4	0.99	0.97	1.00
$m = \frac{n}{100}$	Method 1	0.96	1.00	1.00
	Method 2	0.95	1.00	1.00
	Method 3	0.99	0.98	0.57
	Method 4	0.99	0.98	1.00

Table 5.3: Frequencies that θ_0 is inside of the credible ball for the squared exponential Gaussian process prior in a non-adaptive setting at given points $x \in \{0.5, 0.41148, 0.31143\}$. Benchmark: Non-distributed method. Method 1: Random partitioning, Method 2: Spatial partitioning, Method 3: Spatial partitioning with inverse variance weights, Method 4: Spatial partitioning with exponential weights. From left to right the sample size is $n = 1000, 4500, 10000$.

linearly with the quantity of data available, the posterior means of the distributed methods concentrates at a sub-optimal rate. On the other hand, we can notice on Table 5.1b that the radius of the L_2 credible ball for every methods is bigger than the L_2 distance between the posterior mean and the true function on average. Furthermore, Tables 5.2 and 5.3 corroborate this statement by showcasing that the coverage obtained by distributed methods in a non-adaptive setting is as good as in the non-

	$n =$	1000	4500	10000
$m = 10$	Benchmark	**1.9800 sec**	**16.5773 sec**	**70.5413 sec**
	Random	1.8536 sec	8.2083 sec	18.3162 sec
	Spatial	1.8900 sec	8.1137 sec	18.3614 sec
$m \approx \sqrt[3]{n}$	Random	1.8536 sec	8.8279 sec	18.1707 sec
	Spatial	1.8900 sec	8.6763 sec	18.5524 sec
$m = \frac{n}{100}$	Random	1.8536 sec	8.5768 sec	18.5996 sec
	Spatial	1.8900 sec	8.3238 sec	18.4250 sec

Table 5.4: Average running time for the computation of the posterior for θ_0 for the squared exponential Gaussian process prior in a non-adaptive setting. Benchmark: Non-distributed method. Method 1: Random partitioning, Method 2: Spatial partitioning. From left to right the sample size is $n = 1000, 4500, 10000$

distributed case. It is also noted that **Method 2** and **4** may lead to better point-wise coverage than using a non-distributed regression method. This may be due to the capacity of spatially distributed regression to capture localized properties of the "truth".

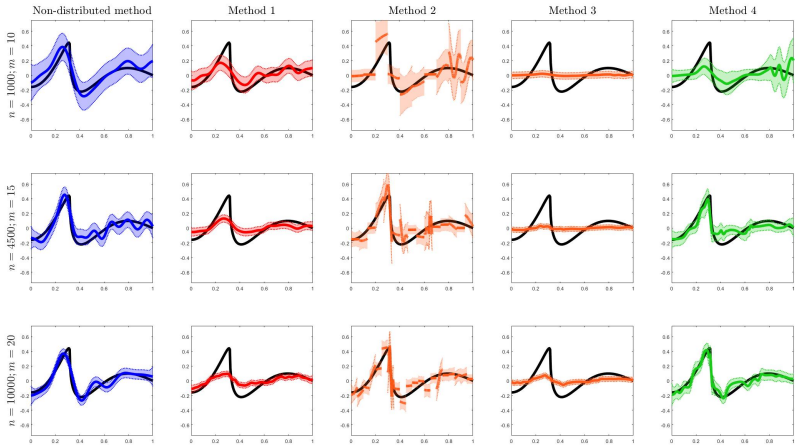

Figure 5.2: Adaptive posterior density for the function θ_0 (drawn in black) on $x \in [0, 1]$. The posterior means are drawn by solid line, while the 95% point-wise credible sets are shaded between two dotted lines. In the first column, we plot the non-distributed method, in the second column the distributed method with random partitioning, the third column the distributed method with spatial partitioning, while in the fourth and fifth column we plot the distributed method with spatial partitioning with inverse variance weights and exponential weights respectively. From top to bottom the sample size is $n = 1000, 5000, 10000$ and the number of experts is $m = 10, 15, 20$.

Although the last-mentioned empirical results draw an optimistic picture of distributed methods, these results are only valid when the exact smoothness β of the function θ_0 is known in advance. In real life applications, this might not be realistic and the smoothness is generally learned before doing any inference on the true function. One can for example estimate the Gaussian process prior parameter from the data using a frequentist technique. We will be using the maximum marginal likelihood estimator (MMLE) in the present study. Despite exhibiting reasonably good contraction rates, the Gaussian process with squared exponential kernel where the

	$n =$	1000	4500	10000
$m = 10$	BM	**32.1979**	**20.2424**	**16.0613**
	M 1	43.3699	34.1139	18.4541
	M 2	41.1122	22.3420	15.6854
	M 3	46.9816	41.1088	33.6787
	M 4	36.6261	21.9179	15.9172
$m \approx n^{1/3}$	M 1	43.3699	38.6536	36.6079
	M 2	41.1122	24.2863	17.9899
	M 3	46.9816	44.6756	43.7188
	M 4	36.6261	22.5384	15.6955
$m = \frac{n}{100}$	M 1	43.3699	45.1984	46.0332
	M 2	41.1122	33.8245	32.4338
	M 3	46.9816	47.9000	48.2887
	M 4	36.6261	27.5078	26.9132

(a) Average L_2 distance between θ_0 and the posterior mean for the squared exponential Gaussian process prior in an adaptive setting.

	$n =$	1000	4500	10000
$m = 10$	BM	**46.6420**	**27.5647**	**20.4758**
	M 1	40.9491	21.6266	17.2895
	M 2	62.1316	39.4716	33.1512
	M 3	23.6971	17.5376	18.1167
	M 4	42.4518	30.6414	27.0113
$m \approx n^{1/3}$	M 1	40.9491	19.4399	14.3160
	M 2	62.1316	43.3218	38.5396
	M 3	23.6971	15.0748	15.0664
	M 4	42.4518	32.0551	28.8076
$m = \frac{n}{100}$	M 1	40.9491	26.2542	27.0996
	M 2	62.1316	55.1480	46.9959
	M 3	23.6971	11.1716	16.3951
	M 4	42.4518	37.2899	38.5320

(b) Average L_2 credible ball radius for the squared exponential Gaussian process prior in an adaptive setting.

Table 5.5: BM: Benchmark, Non-distributed method. M 1: Random partitioning, M 2: Spatial partitioning, M 3: Spatial partitioning with inverse variance weights, M 4: Spatial partitioning with exponential weights. From left to right the sample size is $n = 1000, 4500, 10000$.

		$x = 0.5$			$x = 0.41148$			$x = 0.31143$		
	$n =$	1000	4500	10000	1000	4500	10000	1000	4500	10000
$m = 10$	Benchmark	**0.96**	**0.97**	**0.96**	**0.79**	**0.89**	**0.96**	**0.14**	**0.04**	**0.01**
	Method 1	0.61	0.68	0.89	0.34	0.51	0.79	0.11	0.02	0.00
	Method 2	0.84	0.96	0.93	0.77	0.90	0.98	0.47	0.65	0.42
	Method 3	0.00	0.01	0.08	0.00	0.00	0.00	0.00	0.00	0.00
	Method 4	0.62	0.74	0.83	0.59	0.85	0.96	0.21	0.34	0.24
$m \approx \sqrt[3]{n}$	Method 1	0.61	0.48	0.23	0.34	0.12	0.03	0.11	0.00	0.00
	Method 2	0.84	0.91	0.94	0.77	0.92	0.95	0.47	0.63	0.67
	Method 3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Method 4	0.62	0.73	0.84	0.59	0.83	0.95	0.21	0.17	0.36
$m = \frac{n}{100}$	Method 1	0.61	0.09	0.11	0.34	0.06	0.05	0.11	0.03	0.02
	Method 2	0.84	0.64	0.73	0.77	0.75	0.79	0.47	0.72	0.92
	Method 3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Method 4	0.62	0.43	0.54	0.59	0.72	0.64	0.21	0.32	0.32

Table 5.6: Frequencies that $\theta_0(x)$ is inside of the corresponding credible interval for the squared exponential Gaussian process prior in an adaptive setting at given points $x \in \{0.5, 0.41148, 0.31143\}$. Benchmark: Non-distributed method. Method 1: Random partitioning, Method 2: Spatial partitioning, Method 3: Spatial partitioning with inverse variance weights, Method 4: Spatial partitioning with exponential weights. From left to right the sample size is $n = 1000, 4500, 10000$.

rescaling parameter is estimated using the MMLE is known to provide unreliable uncertainty coverage since the credible sets based thereon fail to cover the true function, see Chapter 2. We will witness this pattern in the following results, in particular in Table 2.7.

In Figure 5.2, one can observe that as soon as one considers adaptation, the

	$N =$	1000	4500	10000
$m = 10$	Benchmark	**0.87**	**0.79**	**0.60**
	Method 1	0.33	0.30	0.38
	Method 2	0.98	0.99	0.99
	Method 3	0.09	0.00	0.02
	Method 4	0.72	0.88	0.93
$m \approx \sqrt[3]{n}$	Method 1	0.33	0.05	0.01
	Method 2	0.98	0.98	1.00
	Method 3	0.09	0.00	0.02
	Method 4	0.72	0.90	0.98
$m = \frac{n}{100}$	Method 1	0.33	0.13	0.15
	Method 2	0.98	1.00	1.00
	Method 3	0.09	0.00	0.09
	Method 4	0.72	0.95	0.99

Table 5.7: Frequencies that θ_0 is inside of the credible ball for the squared exponential Gaussian process prior in an adaptive setting at given points $x \in \{0.5, 0.41148, 0.31143\}$. Benchmark: Non-distributed method. Method 1: Random partitioning, Method 2: Spatial partitioning, Method 3: Spatial partitioning with inverse variance weights, Method 4: Spatial partitioning with exponential weights. From left to right the sample size is $n = 1000, 4500, 10000$.

	$n =$	1000	4500	10000
$m = 10$	Benchmark	**8.0086 sec**	**3.40244 min**	**23.81 min**
	Random	3.7013 sec	20.6797 sec	79.5956 sec
	Spatial	3.6724 sec	21.5390 sec	81.8753 sec
$m \approx \sqrt[3]{n}$	Random	3.7013 sec	17.5368 sec	49.5480 sec
	Spatial	3.6724 sec	16.36173 sec	49.1387 sec
$m = \frac{n}{100}$	Random	3.7013 sec	16.4724 sec	29.9921 sec
	Spatial	3.6724 sec	15.0549 sec	32.0985 sec

Table 5.8: Average running time for the computation of the posterior for θ_0 for the squared exponential Gaussian process prior in an adaptive setting. Benchmark: Non-distributed method. From left to right the sample size is $n = 1000, 4500, 10000$

distributed methods do not behave similarly any longer. Indeed, the way the data is partitioned influences how well the true function can be recovered with the global posteriors. For instance, if we randomly partition the data across the machines and the draw local Bayesian inference using local empirical Bayes estimates, the global posterior will not contract optimally around the true function. This phenomenon has already been studied theoretically in (Szabó and van Zanten, 2019) in the signal-in-white-noise model. Using spatial partitioning could seem more sensible since each machine could adapt to the smoothness locally on a smaller region. This method does result in a global posterior which behaves similarly to the non-distributed method, and yet the discontinuities it inherently produces make it unattractive to practitioners. Fortunately, putting appropriate weights on each draw of the local posteriors, namely $\omega_k(x) = W(x)w_k(x)$, where $w_k(x) = e^{-m^2(x-c_k)^2}/\sigma_k^2(x)$ with c_k being the center of the local data region and $W(x) = \left(\sum_{j=1}^m w_k(x)\right)^{-1}$, alleviate the issue of the "jumps" at the border of the different partitioning regions while preserving the optimal recovery property of the method.

Tables 5.5a, 5.5b, 5.6 and 5.7 support the conclusion drawn from Figures 5.2. They illustrate that if one wants to adapt the priors to the smoothness locally before computing the global posterior, then the distributed method of choice influences greatly the performance of the corresponding posterior. Some methods (**Methods 1** and **3**) result in very poor performances, which can be exacerbated if the number of experts increases with the data. Considering this dysfunction, it may seem that distributed adaptation sometimes leads to a global posterior behaving as bad as the worst local posterior in terms of contraction around the true function and credible set coverage. On the other hand, the other methods (**Methods 2** and **4**) exhibit promising results when that $m \lesssim \sqrt[3]{n}$. Not only are the rates at which the posterior means approach the true function for those methods on par with their non-distributed counterpart, both the point-wise and the L_2 coverage are sometimes slightly improved due to the localized aspect of the former methods. Moreover, we can notice that the coverage is still good even when the number of machines increases linearly with n, which indicates that even when the methods do not achieve optimal recovery, the global posterior does not concentrate too much around the global posterior mean. It should be taken into account that these results are nonetheless only numerical.

While Table 5.4 highlighted the gain in computation times distributed methods offer when the smoothness of the true function is correctly assumed, Table 5.8 emphasizes that this gain is considerable in an adaptive setting. As a matter of fact, the computation of the MMLE is also heavily influenced by the size of the data which explains why the computation of distributed methods takes much less time than the computation of the classical posterior.

Overall, most methods are of interest, especially when the smoothness is assumed to be known, although some of them perform sub-optimally in the adaptive setting. It seems that spatial partitioning with exponentially decreasing weights is the method that generates a global posterior closest to the long-established conventional posterior when the number of experts increase reasonably fast.

§5.2.2 Airline Delays (USA Flight)

Next, we will compare the performance of our different distributive method on a large-scale data set: flight arrival and departure times for every commercial flight in the USA from January 2008 to April 2008. This data set covers more than 5 million flights and contains exhaustive information about the flights, including delays at arrival (in minutes). The average delay in first the quarter of 2008 was about 30 minutes, but one may be interested in estimating this delay more accurately thanks to wealth of data available. However, the usual non-parametric Bayesian regression is discouraged due to the mere size of the data set

This data set has already been studied before by (Hensman et al., 2013), (Gal et al., 2014), (Ng and Deisenroth, 2014) and (Ng and Deisenroth, 2015) using various methods to speed-up the regression. (Hensman et al., 2013) used Stochastic Variational inference (SVI) with inducing points, whereas (Ng and Deisenroth, 2015) compared different distributed methods with random partitioning, among which their robust Bayesian Committee Machine (rBCM) performed the best. We decided to follow the same procedure described in those articles; in order to predict the delay at arrival we select $P = 70K, 2M$ and $5M$ data points to train our models and $100,000$

$P =$	$70K$			$2M$			$5M$		
	CT	RMSE	SE	CT	RMSE	SE	CT	RMSE	SE
SVI	–	33.0	–	–	–	–	–	–	–
rBCM	13 s	27.1	< 0.3	39 s	34.4	< 0.3	90 s	35.5	< 0.3
M 1	24.2 m	29.1	0.52	35.4 m	34.8	0.49	41.3 m	41.5	0.5
M 2	22.5 m	25.0	0.12	31.1 m	27.1	0.14	39.9 m	30.2	0.17
M 3	24.6 m	33.4	0.35	35.0 m	40.4	0.29	40.8 m	45.1	0.28
M 4	23.7 m	26.6	0.15	35.5 m	31.5	0.14	42.1 m	31.8	0.15

Table 5.9: US Flight Data Set. Performance of different method in terms of computation time (CT), root-mean-square error (RMSE) and standard error (SE). SVI and rBCM results are reported from (Hensman et al., 2013) and (Ng and Deisenroth, 2015) respectively. Best and worse performance by training size are highlighted in blue and red, respectively. M 1: Random partitioning, M 2: Spatial partitioning, M 3: Spatial partitioning with inverse variance weights, M 4: Spatial partitioning with exponential weights.

other data points to test them. The dependent variable is of dimension 8 and encompass: the age of the aircraft (number of years since deployment), distance between the two airports (in miles), airtime (in minutes), departure time, arrival time, month, day of the week and day of the month. We conducted 10 experiments with 256, 512 and 1024 machines respectively. The computation times, the root-mean-square errors and the standard errors (i.e. root of the mean of the squares of the deviations within the training set) of the different methods are all reported in Table 5.9, along with the reported performance of the SVI and the rBCM. All the simulation have been made on a single workstation using an Intel Core i7-8700 CPU operating at 3.40 GHz and 16 GB of RAM using sequential computation of the different local GP posteriors (i.e. all the local posteriors have been computed and stored on the same station).

On the table, one can observe a decrease in performances with the number of training data which has already been reported in Ng and Deisenroth (2015). Nonetheless, it is also noticeable that partitioning randomly the data across machines leads to similar RMSE as the one obtaining by an rBCM, which is not surprising. It is noticeable that the spatial partitioning consistently outperforms all the other GP methods. Nonetheless, we should remind the reader that despite those performance, the resulting global posterior contains multiple discontinuity regions. The table draws however a dark picture on the prediction of weighting the local posterior with inverse variance after a spatial partitioning of the data. Luckily, this can be compensated by exponential weights which largely improve the prediction. Indeed, **Method 4** achieves consistently better RMSE than other reported methods in the literature while still providing a global posterior with continuous draws.