Direct Loss Minimization for Sparse Gaussian Processes: Supplementary Materials

Yadi Wei
Indiana University

Rishit Sheth
Microsoft Research New England

Roni Khardon
Indiana University

1 Efficient Implementation of Product Sampling

Efficient Rejection Sampling: Recall that we want to sample from $\tilde{q}(f|\theta) = \frac{q(f|\theta)p(y|f)}{\mathrm{E}_{q(f|\theta)}\, p(y|f)}$ where the normalizing constant $\mathrm{E}_{q(f|\theta)}\, p(y|f)$ is not known. Naive rejection sampling will have a high rejection rate and more advanced sampling techniques, such as adaptive rejection sampling, will be too slow because we need to sample the gradient for each example in each minibatch of optimization. We next show how to take advantage of the structure of $\tilde{q}(f)$ to construct an efficient sampler. Recall the standard setting for rejection sampling. To sample from an unnormalized distribution $h_1(f)$ we introduce $h_2(f)$ which is easy to sample from and such that $Kh_2(f) \geq h_1(f)$. Then we sample $f^* \sim h_2(f)$, and accept f^* with probability $h_1(f^*)/Kh_2(f^*)$.

In our case h_1 is a product of a normal distribution $q(f) = \mathcal{N}(\mu, \sigma^2)$ and a likelihood function $\ell(f) = p(y|f)$. In the following we assume that $\ell(f) \leq \ell_{max}$ is bounded, which is true for discrete y and can be enforced by lower bounding the variance when y is continuous. The main issue for sampling is the overlap between the "high value regions" of $q()$ and $\ell()$. If they are well aligned, for example, $\mathrm{argmax}_{f\in\mu\pm\sigma}\ell(f) \geq 0.5$, then we can use $h_2(f) = q(f)$ with $K = 1$ and the rejection rate will not be high. However, if they are not aligned then sampling from $q()$ will have a high rejection rate. To address this, we fix a small integer n and sample from a broader distribution with the same mean $h_2(f) = \mathcal{N}(\mu, n\sigma^2)$.

Let a, b be the intersection points of the PDFs of $q()$ and $h_2()$ ($\mu \pm r$ for $r = \sigma\sqrt{\log n/(1-1/n)}$) and let $m_1 = \max_{f\in[a,b]}\ell(f)$ and $m_2 = \min_{f\in[a,b]}\frac{h_2(f)}{q(f)} = \frac{1}{\sqrt{n}}$. Note that $\frac{m_1}{m_2}$ increases with n. To balance the sampling ratios within and outside $[a, b]$, we pick the largest $n \leq 10$ s.t. $m_1 \leq m_2\ell_{max}$ and use $K = \ell_{max}$. Then in the interval $[a, b]$ we have $h_2(f)\ell_{max} \geq h_2(f)\frac{m_1}{m_2} \geq q(f)\ell(f)$ and outside the interval we have $h_2(f) \geq q(f)$ and therefore $h_2(f)\ell_{max} \geq q(f)\ell(f)$ as required.

The only likelihood specific step in the computation is the value of m_1. For the binary case with sigmoid or probit likelihood the maximum is obtained at one of the endpoints $p(a), p(b)$. For count regression with Poisson likelihood with link function $\lambda = e^f$, if the observation $\log y \in [a, b]$ then we also need to evaluate $p(y|\lambda = y)$. The crucial point is that because of the structure of $q()$ and $h_2()$ the values of m_1,m_2 can be calculated analytically in constant time and the cost of determining n is not prohibitive.

Vectorized sampling: The process above yields efficient sampling, where after an initial set of learning iterations the average number of rejected samples is low (approximately 2 in our evaluation). However, in practice the process is still slow. One of the reasons is the fact that we calculate n which defines the sampling distribution separately for each example i and then perform rejection sampling separately for each i. Modern implementations gain significant speedup by vectorizing operations, but this is at odds with individual rejection sampling. We partly alleviate this cost by a hybrid procedure as follows. Note that for each i we have $h_2(f_i) = \mathcal{N}(\mu_i, n_i\sigma_i^2)$ and that the samples for different i's are independent. We can therefore collect these and sample from a multivariate normal with diagonal covariance. However, each such vector of samples will have some rejected entries. Our hybrid procedure repeats the vectorized sampling twice, uses the first successful sample for each i, and for entries which had no successful sample, resorts to individual sampling. We have found that this reduces overall run time by at least 50%.

2 Convergence of smooth-bMC with Probability 1

This section develops an analysis of smoothed bMC estimates. It is shown that by adding a small factor to the denominator of the gradient estimate we can bound the step direction and guarantee convergence w.p. 1.

Consider gradient based minimization of a function $f()$ with step direction $g_t = s_t + w_t$ and step size γ_t.

Assumption 7. *The function $f : \mathcal{R}^n \to \mathcal{R}$, and s_t, w_t, γ_t satisfy the following conditions:*

(a) $f(r) \geq 0$ *for all* $r \in \mathcal{R}^n$.

(b) The function f is continuously differentiable and there exists some constant L such that

$$\|\nabla f(r) - \nabla f(\bar{r})\| \leq L\|r - \bar{r}\|, \forall r, \bar{r} \in \mathcal{R}^n.$$

(c) There exists positive constant c_1, c_2 such that $\forall t$,

$$c_1\|\nabla f(r_t)\|^2 \leq -\nabla f(r_t)^T \operatorname{E}[s_t|\mathcal{F}_t],$$
$$\operatorname{E}[\|s_t\|^2] \leq c_2\|\nabla f(r_t)\|^2.$$

(d) There exists positive constant p, q such that

$$\operatorname{E}[\|w_t\|^2] \leq (\gamma_t(q + p\|\nabla f(r_t)\|))^2.$$

Notice that condition (d) in Assumption 7 implies $\operatorname{E}[\|w_t\|] \leq \gamma_t(q+p\|\nabla f(r_t)\|)$. This can be derived from Jensen's inequality where the quadratic function is convex.

Proposition 8. *Consider the algorithm*

$$r_{t+1} = r_t + \gamma_t(s_t + w_t),$$

where the stepsizes γ_t are nonnegative and satisfy

$$\sum_{t=0}^{\infty} \gamma_t = \infty, \sum_{t=0}^{\infty} \gamma_t^2 < \infty.$$

Under Assumption 7, the following hold with probability 1:

(a) The sequence $f(r_t)$ converges.

(b) We have $\lim_{t\to\infty} \nabla f(r_t) = 0$.

(c) Every limit point of r_t is a stationary point of f.

The proposition and its proof are a slight modification of Proposition 4.1 by Bertsekas and Tsitsiklis (1996). Compared to that result, Assumption 7 splits the conditions on the step direction $g_t = s_t + w_t$ from Bertsekas and Tsitsiklis (1996) into portion c on s_t and portion d on w_t. This slight weakening of the condition enables our application in Corollary 9. We include the proof here for completeness.

Proof. This proof slightly modifies the proof of Proposition 4.1 in Bertsekas and Tsitsiklis (1996). As shown there (in Eq 3.39), if $\nabla f()$ is L-Lipschitz then, for two vectors r, z, $f(r+z) - f(r) \leq z^T\nabla f(r) + \frac{L}{2}\|z\|^2$. Then replacing z with $\gamma_t(s_t + w_t)$ and taking expectation, we have

$$\begin{aligned}
\operatorname{E}[f(r_{t+1})] &\leq f(r_t) + \gamma_t\nabla f(r_t)^T \operatorname{E}[s_t + w_t] + \frac{\gamma_t^2 L}{2}\operatorname{E}[\|s_t + w_t\|^2] \\
&\leq f(r_t) + \gamma_t\nabla f(r_t)^T \operatorname{E}[s_t] + \gamma_t\nabla f(r_t)^T \operatorname{E}[w_t] + \gamma_t^2 L \operatorname{E}[\|s_t\|^2] + \gamma_t^2 L \operatorname{E}[\|w_t\|^2] \\
&\leq f(r_t) + \gamma_t(-c_1\|\nabla f(r_t)\|^2 + \|\nabla f(r_t)\| \operatorname{E}[\|w_t\|]) + \gamma_t^2 L(c_2^2\|\nabla f(r_t)\|^2 \\
&\quad + \gamma_t^2 q^2 + 2\gamma_t^2 pq\|\nabla f(r_t)\| + \gamma_t^2 p^2\|\nabla f(r_t)\|^2) \\
&\leq f(r_t) - \gamma_t(c_1 - \gamma_t p - \gamma_t c_2^2 L - \gamma_t^3 p^2 L)\|\nabla f(r_t)\|^2 + \gamma_t^2(q + 2\gamma_t^2 pqL)\|\nabla f(r_t)\| + \gamma_t^4 q^2 L.
\end{aligned}$$

The second inequality uses $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ for any two vector a, b. The third inequality uses the conditions in Assumption 7. Let $c_t = c_1 - \gamma_t p - \gamma_t c_2^2 L - \gamma_t^3 p^2 L$, $d_t = q + 2\gamma_t^2 pqL$. Then,

$$\begin{aligned}
\mathrm{E}[f(r_{t+1})] &\leq f(r_t) - \gamma_t c_t \|\nabla f(r_t)\|^2 + \gamma_t^2 d_t \|\nabla f(r_t)\| + \gamma_t^4 q^2 L \\
&\leq f(r_t) - \gamma_t c_t \|\nabla f(r_t)\|^2 + \gamma_t^2 d_t (1 + \|\nabla f(r_t)\|^2) + \gamma_t^4 q^2 L \\
&= f(r_t) - \gamma_t (c_t - \gamma_t d_t) \|\nabla f(r_t)\|^2 + \gamma_t^2 d_t + \gamma_t^4 q^2 L \\
&= f(r_t) - X_t + Z_t,
\end{aligned}$$

where $X_t = \begin{cases} \gamma_t(c_t - \gamma_t d_t)\|\nabla f(r_t)\|^2, & \text{if } c_t \geq \gamma_t d_t, \\ 0, & \text{otherwise,} \end{cases}$, and

$Z_t = \begin{cases} \gamma_t^2 d_t + \gamma_t^4 q^2 L, & \text{if } c_t \geq \gamma_t d_t, \\ \gamma_t^2 d_t + \gamma_t^4 q^2 L - \gamma_t(c_t - \gamma_t d_t)\|\nabla f(r_t)\|^2, & \text{otherwise.} \end{cases}$

Notice that $c_t - \gamma_t d_t$ is monotonically decreasing in γ_t and $\lim_{t\to\infty} \gamma_t = 0$, so there exists some finite time after which $\gamma_t d_t \leq c_t$. It follows that after some finite time, we have $Z_t = \gamma_t^2 d_t + \gamma_t^4 q^2 L$ and therefore $\sum_{t=0}^{\infty} Z_t < \infty$. Applying Proposition 4.2 (Supermartingale Convergence Theorem) in Bertsekas and Tsitsiklis (1996), we can conclude that $f(r_t)$ converges which establishes part (a) of the proposition, and in addition that $\sum_t X_t < \infty$.

Similarly after some time, we have $c_t - \gamma_t d_t \geq \frac{c_1}{2}$ and

$$X_t = \gamma_t(c_t - \gamma_t d_t)\|\nabla f(r_t)\|^2 \geq \frac{c_1}{2}\gamma_t \|\nabla f(r_t)\|^2.$$

Hence,

$$\sum_{t=0}^{\infty} \gamma_t \|\nabla f(r_t)\|^2 < \infty.$$

Below we prove that $\|\nabla f(r_t)\|$ converges to 0. Let $g_t = s_t + w_t$,

$$\begin{aligned}
\mathrm{E}[\|g_t\|^2] &= \mathrm{E}[\|s_t + w_t\|^2] \\
&\leq \mathrm{E}[2\|s_t\|^2 + 2\|w_t\|^2] \\
&\leq 2c_2\|\nabla f(r_t)\|^2 + 2\gamma_t^2(q + p\|\nabla f(r_t)\|)^2 \\
&= 2c_2\|\nabla f(r_t)\|^2 + 2\gamma_t^2(q^2 + 2pq\|\nabla f(r_t)\| + p^2\|\nabla f(r_t)\|^2) \\
&\leq 2(c_2 + p^2\gamma_t^2 + 2pq\gamma_t^2)\|\nabla f(r_t)\|^2 + 2\gamma_t^2 q^2 + 4pq\gamma_t^2.
\end{aligned}$$

Suppose $\max \gamma_t \leq \gamma$, let $K_1 = 2(c_2 + p^2\gamma^2 + 2pq\gamma^2)$ and $K_2 = 2\gamma^2 q^2 + 4pq\gamma^2$, we have $\mathrm{E}[\|g_t\|^2] \leq K_1\|\nabla f(r_t)\|^2 + K_2$. Then all the remaining steps in the proof in (Bertsekas and Tsitsiklis, 1996) for claims (b),(c) can be followed by replacing s_t in Bertsekas and Tsitsiklis (1996) with our g_t. □

In order to establish convergence w.p. 1 we need to bound the norm of the step direction. To achieve this we add a smoothing parameter ν to the denominator of the estimate. This yields the *smooth-bMC* algorithm, whose step directions are

$$\begin{aligned}
d_{i,m}(r) &:= \frac{\sum_{\ell=1}^{L} \phi_i'(r, \epsilon^{(\ell)})}{\sum_{\ell=1}^{L} \phi_i(r, \epsilon^{(\ell)}) + \nu} a_{i,1} \\
d_{i,V}(r) &:= \frac{\sum_{\ell=1}^{L} \phi_i''(r, \epsilon^{(\ell)})}{\sum_{\ell=1}^{L} \phi_i(r, \epsilon^{(\ell)}) + \nu} \frac{a_{i,2}a_{i,2}^\top}{2}
\end{aligned}$$

and thus

$$w_{t,i,m} = \frac{a_{i,1}}{n}\left(\frac{(1/L)\sum_\ell \phi_i'(r_t, \epsilon^{(\ell)})}{(1/L)\sum_\ell \phi_i(r_t, \epsilon^{(\ell)}) + \nu_t} - \frac{\mathrm{E}_{\mathcal{N}(\epsilon|0,1)}\, \phi_i'(r_t, \epsilon)}{\mathrm{E}_{\mathcal{N}(\epsilon|0,1)}\, \phi_i(r_t, \epsilon)}\right)$$

and a similar expression holds for V's portion. We now have:

Corollary 9. *If for every t and i, $\mathrm{E}_{q(f_i|r)}\, p(y_i|f_i) \geq \zeta > 0$, smooth-bMC uses $\nu_t = \gamma_t \zeta$, and*

$$L > \frac{\log(6n/\delta_t)}{2\gamma_t^2} \max\left\{ \frac{B^2}{|\,\mathrm{E}_{\mathcal{N}(\epsilon|0,1)}\, \phi_i(r,\epsilon)|^2}, \frac{(B'-b')^2}{P^2}, \frac{(B''-b'')^2}{Q^2} \right\}, \tag{18}$$

where $P = \begin{cases} |\,\mathrm{E}_{\mathcal{N}(\epsilon|0,1)}\, \phi_i'(r,\epsilon)|, & \text{if } \mathrm{E}_{\mathcal{N}(\epsilon|0,1)}\, \phi_i'(r,\epsilon) \neq 0 \\ 1, & \text{otherwise} \end{cases}$, $Q = \begin{cases} |\,\mathrm{E}_{\mathcal{N}(\epsilon|0,1)}\, \phi_i''(r,\epsilon)|, & \text{if } \mathrm{E}_{\mathcal{N}(\epsilon|0,1)}\, \phi_i''(r,\epsilon) \neq 0 \\ 1, & \text{otherwise} \end{cases}$,
and $\delta_t = \gamma_t^4$, then smooth-bMC satisfies the conditions of Proposition 8 and hence converges w.p. 1.

Proof. Conditions (a,b,c) of Assumption 7 are handled exactly as in the main paper. Thus we only need to show that (d) holds.

First consider the case when $\mathrm{E}[\phi'] \neq 0$ and $\mathrm{E}[\phi''] \neq 0$. With

$$L \geq \frac{\log(6n/\delta_t)}{2\alpha^2} \max\left\{ \frac{B^2}{\mathrm{E}[\phi_i(r,\epsilon)]^2}, \frac{(B'-b')^2}{|\mathrm{E}[\phi_i'(r,\epsilon)]|^2}, \frac{(B''-b'')^2}{|\mathrm{E}[\phi_i''(r,\epsilon)]|^2} \right\}$$

we have that $\forall i, t$,

$$(1-\alpha)|\,\mathrm{E}[\phi_i]| \leq \left| (1/L) \sum_{l=1}^{L} \phi_i(r,\epsilon^{(l)}) \right| \leq (1+\alpha)|\,\mathrm{E}[\phi_i]|,$$
$$(1-\alpha)|\,\mathrm{E}[\phi_i']| \leq \left| (1/L) \sum_{l=1}^{L} \phi_i'(r,\epsilon^{(l)}) \right| \leq (1+\alpha)|\,\mathrm{E}[\phi_i']|,$$
$$(1-\alpha)|\,\mathrm{E}[\phi_i'']| \leq \left| (1/L) \sum_{l=1}^{L} \phi_i''(r,\epsilon^{(l)}) \right| \leq (1+\alpha)|\,\mathrm{E}[\phi_i'']|.$$

hold simultaneously w.p. $\geq 1 - \frac{\delta_t}{n}$.

Since $\phi_i > 0$, we have $(1/L)\sum_{l=1}^{L} \phi_i(r,\epsilon^{(l)}) + \nu \geq (1-\alpha)\,\mathrm{E}[\phi_i] + \nu \geq (1-\alpha)\,\mathrm{E}[\phi_i]$. In addition

$$(1/L)\sum_{l=1}^{L} \phi_i(r,\epsilon^{(l)}) + \nu \leq (1+\alpha)\,\mathrm{E}[\phi_i] + \nu \leq (1+\alpha+\frac{\nu}{\zeta})\,\mathrm{E}[\phi_i].$$

Then

$$\begin{aligned} \|w_{t,i,m}\|^2 &\leq \frac{\|a_{i,1}\|^2}{n^2} \max\left(\frac{1+\alpha}{1-\alpha} - 1, 1 - \frac{1-\alpha}{1+\alpha+\frac{\nu}{\zeta}} \right)^2 \left(\frac{\mathrm{E}[\phi_i'(r_t,\epsilon)]}{\mathrm{E}[\phi_i(r_t,\epsilon)]} \right)^2 \\ &= \frac{\|a_{i,1}\|^2}{n^2} \max\left(\frac{2\alpha}{1-\alpha}, \frac{2\alpha+\frac{\nu}{\zeta}}{1+\alpha+\frac{\nu}{\zeta}} \right)^2 \left(\frac{\mathrm{E}[\phi_i'(r_t,\epsilon)]}{\mathrm{E}[\phi_i(r_t,\epsilon)]} \right)^2. \end{aligned}$$

Let $\nu = \alpha\zeta$, then $\frac{\nu}{\zeta} = \alpha$. Thus

$$\max\left(\frac{2\alpha}{1-\alpha}, \frac{2\alpha+\frac{\nu}{\zeta}}{1+\alpha+\frac{\nu}{\zeta}} \right) = \max\left(\frac{2\alpha}{1-\alpha}, \frac{3\alpha}{1+2\alpha} \right) \leq \frac{3\alpha}{1-\alpha}.$$

Using $\alpha \leq 0.5$ we get $\|w_{t,i,m}\|^2 \leq \left(\frac{6B^*\alpha}{\zeta} \frac{\|a_{i,1}\|}{n} \right)^2$.

Next consider the case when $\mathrm{E}[\phi'] = 0$. In this case we can select $L \geq \frac{(B'-b')^2 \log(6n/\delta_t)}{2\alpha^2}$ such that $|(1/L)\sum_l \phi'(r, \epsilon^{(l)})| \leq \alpha$ w.p.$\geq 1 - \delta_t/(3n)$. At the same time, $(1/L)\sum_l \phi_i(r, \epsilon^{(l)}) + \nu \geq (1-\alpha)\,\mathrm{E}[\phi_i(r, \epsilon)]$. Thus,

$$\|w_{t,i,m}\|^2 \leq \left(\frac{\|a_{i,1}\|}{n}\frac{\alpha}{\zeta(1-\alpha)}\right)^2 \leq \left(\frac{2\alpha}{\zeta}\frac{\|a_{i,1}\|}{n}\right)^2.$$

Combined two cases, we have

$$\|w_{t,i,m}\|^2 \leq \left(\frac{2\alpha}{\zeta}\frac{\|a_{i,1}\|}{n}(3B^*+1)\right)^2.$$

$w_{t,i,V}$ can be bounded similarly. Thus, $\|w_{t,i}\|^2$ can be upper bounded with high probability. Overall, w.p. at least $1-\delta_t$,

$$\|w_t\|^2 \leq \sum_i \sum_j \|w_{t,i}\|\|w_{t,j}\| \leq \left(A\frac{2\alpha}{\zeta}(3B^*+1)\right)^2,$$

where $A = \max_i \|a_i\|$.

However, w.p. at most δ_t, the above inequality does not hold. In order to bound the expectation we use the following upper bound which always holds:

$$\begin{aligned}
\|w_{t,i,m}\|^2 &= \frac{\|a_{i,1}\|^2}{n^2}\left|\frac{(1/L)\sum_l \phi_i'(r,\epsilon^{(l)})}{\nu + (1/L)\sum_l \phi_i(r,\epsilon)} - \frac{\mathrm{E}[\phi_i'(r,\epsilon)]}{\mathrm{E}[\phi_i(r,\epsilon)]}\right|^2 \\
&\leq \frac{\|a_{i,1}\|^2}{n^2}\left(2\left(\frac{(1/L)\sum_l \phi_i'(r,\epsilon^{(l)})}{\nu + (1/L)\sum_l \phi_i(r,\epsilon^{(l)})}\right)^2 + 2\left(\frac{\mathrm{E}[\phi_i'(r,\epsilon)]}{\mathrm{E}[\phi_i(r,\epsilon)]}\right)^2\right) \\
&\leq 2\frac{\|a_{i,1}\|^2}{n^2}\max\{|b'|,|B'|\}^2\left(\frac{1}{\nu^2}+\frac{1}{\zeta^2}\right) \\
&\leq 2\frac{\|a_{i,1}\|^2}{n^2}(B^*)^2\left(\frac{1}{\nu^2}+\frac{1}{\zeta^2}\right) \\
&= 2\frac{\|a_{i,1}\|^2}{n^2}(B^*)^2\left(\frac{1}{\alpha^2}+1\right)\frac{1}{\zeta^2} \\
&\leq 4\frac{\|a_{i,1}\|^2}{n^2}(B^*)^2\frac{1}{\alpha^2\zeta^2}.
\end{aligned}$$

In the third step, we used the fact that ϕ_i' is bounded between b' and B'. In the last step, we use the fact that $1 \leq \frac{1}{\alpha^2}$. Similar arguments can be derived for $w_{t,i,V}$. Thus $\|w_{t,i}\|^2 \leq 4\frac{\|a_i\|^2}{n^2}(B^*)^2\frac{1}{\alpha^2\zeta^2}$. Further,

$$\|w_t\|^2 = \|\sum_i w_{t,i}\|^2 \leq \sum_i\sum_j \|w_{t,i}\|\|w_{t,j}\| \leq 4A^2(B^*)^2\frac{1}{\alpha^2\zeta^2} = \frac{D}{\alpha^2}.$$

where $D = \left(A\frac{2}{\zeta}B^*\right)^2$ is a constant.

Thus, $\mathrm{E}[\|w\|^2]$ can be bounded,

$$\mathrm{E}[\|w_t\|^2] \leq (1-\delta_t)(A\frac{2\alpha}{\zeta}(3B^*+1))^2 + \delta_t D \leq (A\frac{2\alpha}{\zeta}(3B^*+1))^2 + \delta_t\frac{D}{\alpha^2}.$$

Here let $\alpha = \gamma_t$ (hence $\nu_t = \gamma_t\zeta$) and $\delta_t = \gamma_t^4$, then condition (d) of Assumption 7 holds with $p=0$ and $q^2 = (A\frac{2}{\zeta}(3B^*+1))^2 + \left(A\frac{2}{\zeta}B^*\right)^2$. □

Suppose $\gamma_t = \frac{1}{t}$, then the sample size $L \propto t^2\log(nt)$, which matches the sample size in the main paper. As in the discussion there, in practice we use a fixed sample size L and we use a fixed smoothing factor ν.

3 Proof of Proposition 4 from Main Paper: Bounds on ϕ, ϕ', ϕ''

In this section we show that bounds of the form $\phi < B$, $b' < \Phi' < B'$, and $b'' < \Phi'' < B''$ holds in each cases as listed in the following table:

Likelihood	B	b'	B'	b''	B''
Logistic, $\sigma(yf)$	1	$-\frac{1}{4}$	$\frac{1}{4}$	$-\frac{1}{4}$	$\frac{1}{4}$
Gaussian, $e^{-(y-f)^2/2\sigma^2}, c=\frac{1}{\sqrt{2\pi}\sigma}$	c	$-\frac{c}{\sqrt{e}\sigma}$	$\frac{c}{\sqrt{e}\sigma}$	$-\frac{c}{\sigma^2}$	$\frac{2c}{\sigma^2 e^{3/2}}$
Probit, $\Phi(yf)$, Φ is cdf of Gaussian	1	$-1/\sqrt{2\pi}$	$1/\sqrt{2\pi}$	$-1/\sqrt{2\pi e}$	$1/\sqrt{2\pi e}$
Poisson, $\frac{g(f)^y e^{g(f)}}{y!}, g(f)=\log(e^f+1)$	1	-1	1	-2.25	2.25
Poisson, $\frac{g(f)^y e^{g(f)}}{y!}, g(f)=e^f$	1	$-y-1$	y	$-y-1/4$	$2y^2+3y+2$
Student's t, $c(1+\frac{(y-f)^2}{\sigma^2\nu})^{-\frac{\nu+1}{2}}, c=\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu}\sigma}$	c	$-\frac{\frac{c}{\sigma}\frac{\nu+1}{\nu}\sqrt{\frac{\nu}{\nu+2}}}{(\frac{\nu+3}{\nu+2})^{(\nu+3)/2}}$	$\frac{\frac{c}{\sigma}\frac{\nu+1}{\nu}\sqrt{\frac{\nu}{\nu+2}}}{(\frac{\nu+3}{\nu+2})^{(\nu+3)/2}}$	$-\frac{c}{\sigma^2}\frac{\nu+1}{\nu}$	$2\frac{c}{\sigma^2}\frac{\nu+1}{\nu}(\frac{\nu+2}{\nu+5})^{(\nu+5)/2}$

logistic: For convenience let the label y_i be in $\{-1,1\}$. Then $\phi=\sigma(y_i f_i)\le 1$, $\phi'=y_i\sigma(y_i f_i)(1-\sigma(y_i f_i))\in[-0.25,0.25]$, and $\phi''=y_i^2\sigma(y_i f_i)(1-\sigma(y_i f_i))[1-2\sigma(y_i f_i)]\in[-0.25,0.25]$.

Gaussian: The Gaussian likelihood is $\phi=p(y|f,\sigma^2)=c\exp(-(y-f)^2/2\sigma^2), c=\frac{1}{\sqrt{2\pi}\sigma}$. In the following, let $x=(y-f)/\sigma$ to reduce clutter. The first derivative of ϕ w.r.t. f is given by $\phi'=c\exp(-x^2/2)x\frac{1}{\sigma}$ whose root at $f=y$ ($x=0$) corresponds to the maximum of the likelihood c. The second derivative is given by $\phi''=\frac{c}{\sigma^2}\exp(-x^2/2)(x^2-1)$. The first derivative evaluations at the second derivative roots defined by $f=y\pm\sigma$ ($x=\pm1$) are $\pm\frac{c}{\sigma}\sqrt{e}$. The third derivative is given by $\phi'''=\frac{c}{\sigma^3}\exp(-x^2/2)x(x^2-3)$. The second derivative evaluations at the third derivative roots defined by $f=y$ ($x=0$) and $f=y\pm\sigma\sqrt{3}$ ($x^2=3$) are $-\frac{c}{\sigma^2}$ and $\frac{2c}{\sigma^2}\exp(-3/2)$, respectively. Finally, the first and second derivatives clearly approach 0 as f approaches $\pm\infty$ since $\exp(f^2/2)$ dominates the growth of any polynomial in f.

probit: For convenience let the label y_i be in $\{-1,1\}$. Then $\phi=\Phi(y_i f_i)$, where Φ is the CDF of the standard normal and clearly $\phi\in[0,1]$. Let $h()$ be the PDF of the standard normal. Then $\phi'=y_i h(y_i f_i)$ and $\phi'=y_i^2 h'(y_i f_i)$. Bounds for these are given by bounds above for the normal distribution with $\mu=0$ and $\sigma^2=1$, where we have to account for the sign flip in $\phi'=y_i h()$.

Poisson: Here we also need to consider the link function. Two standard options are $\lambda=g(f_i)=e^{f_i}$ or $\lambda=g(f_i)=\ln(e^{f_i}+1)$.

For the first option we have $\lambda=g(f_i)=e^{f_i}$. As above it is obvious that $\phi\le 1$. $\phi'=\frac{\lambda^y e^{-\lambda}}{y!}(y-\lambda)=y\phi-(y+1)\frac{\lambda^{y+1}e^{-\lambda}}{(y+1)!}$. Thus, $\phi'\ge b'=-y-1$ and $\phi'\le B'=y$. $\phi''=\frac{\lambda^y e^{-\lambda}}{y!}(y^2-(2y+1)\lambda+\lambda^2)=\frac{\lambda^y e^{-\lambda}}{y!}((\lambda-(y+\frac{1}{2}))^2-y-\frac{1}{4})\ge -y-\frac{1}{4}$. On the other hand, $\phi''=(\frac{\lambda^y e^{-y}}{y!})y^2-\frac{\lambda^{y+1}e^{-\lambda}}{(y+1)!}(2y+1)(y+1)+\frac{\lambda^{y+2}e^{-\lambda}}{(y+2)!}(y+1)(y+2)\le 2y^2+3y+2$.

For the second option we have $\lambda=g(f_i)=\ln(e^{f_i}+1)$. Below we use $\phi_t(\lambda_i)$ to denote $p(t|f_i)=\frac{\lambda_i^t e^{\lambda_i}}{t!}$. First note that $g'(f_i)=\frac{e^{f_i}}{e^{f_i}+1}=\sigma(f_i)\in[0,1]$ and $g''(f_i)\in[0,0.25]$. We have $\phi=\frac{\lambda_i^{y_i}e^{\lambda_i}}{y_i!}\le 1$, $\phi'=g'(f_i)\phi'(\lambda_i)=g'(f_i)(\phi(\lambda_i)-\phi_{y_i-1}(\lambda_i))$ implying $-1<\phi'<1$, and

$$\begin{aligned}\phi'' &= g''(f_i)(\phi(\lambda_i)-\phi_{y_i-1}(\lambda_i))+(g'(f_i))^2(\phi'_{y_i-1}(\lambda_i)-\phi(\lambda_i)')\\ &= g''(f_i)(\phi(\lambda_i)-\phi_{y_i-1}(\lambda_i))+(g'(f_i))^2(\phi_{y_i-2}(\lambda_i)-\phi_{y_i-1}(\lambda_i)-\phi_{y_i-1}(\lambda_i)+\phi_{y_i}(\lambda_i))\end{aligned}$$

which is bounded because all its components are bounded. Plugging in $0\le g'\le 1$, $0\le g''\le 0.25$, we get that $\phi''>-2.25$ and $\phi''<2.25$.

Student T: The student's t likelihood is $p(y|f,\nu,\sigma^2)=c\Big(1+\frac{(y-f)^2}{\sigma^2\nu}\Big)^{-(\nu+1)/2}, c=\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu}\sigma}, \nu\in\mathbb{R}^+$. In the following, let $x=(y-f)/\sigma$ to reduce clutter. The first derivative is given by $-c\frac{\nu+1}{\nu}\frac{x}{(1+\frac{x^2}{\nu})^{(\nu+3)/2}}(-\frac{1}{\sigma})$. The only first derivative root at $f=y$ ($x=0$) corresponds to the maximum of the likelihood c. The second derivative is

given by

$$-c\frac{\nu+1}{\nu}[\frac{1}{(1+\frac{x^2}{\nu})^{(\nu+3)/2}}-\frac{x^2\frac{\nu+3}{\nu}}{(1+\frac{x^2}{\nu})^{(\nu+5)/2}}]\frac{1}{\sigma^2}$$
$$=-c\frac{\nu+1}{\nu}(1+\frac{x^2}{\nu})^{-(\nu+5)/2}[1+\frac{x^2}{\nu}-x^2\frac{\nu+3}{\nu}]\frac{1}{\sigma^2}$$
$$=-c\frac{\nu+1}{\nu}\frac{1-x^2\frac{\nu+2}{\nu}}{(1+\frac{x^2}{\nu})^{(\nu+5)/2}}\frac{1}{\sigma^2}.$$

The first derivative evaluations at the second derivative roots defined by $f=y\pm\sigma\sqrt{\frac{\nu}{\nu+2}}$ ($x=\pm\sqrt{\frac{\nu}{\nu+2}}$) are $\pm\frac{c}{\sigma}\frac{\nu+1}{\nu}\sqrt{\frac{\nu}{\nu+2}}/(\frac{\nu+3}{\nu+2})^{(\nu+3)/2}$. Also, since $\nu>0$, the denominator of the first derivative is a polynomial of degree at least 3 implying that as f approaches $\pm\infty$, the first derivative approaches 0. Hence, the first derivative is bounded over its domain. The third derivative is given by

$$-c\frac{\nu+1}{\nu}[-\frac{\nu+5}{\nu}\frac{x}{(1+\frac{x^2}{\nu})^{(\nu+7)/2}}[1-x^2\frac{2+\nu}{\nu}]+\frac{1}{(1+\frac{x^2}{\nu})^{(\nu+5)/2}}(-2x)\frac{2+\nu}{\nu}](\frac{-1}{\sigma^3})$$
$$=c\frac{\nu+1}{\nu^2}(1+\frac{x^2}{\nu})^{-(\nu+7)/2}x[(\nu+5)[1-x^2\frac{2+\nu}{\nu}]+(1+\frac{x^2}{\nu})2(2+\nu)](\frac{-1}{\sigma^3})$$
$$=c\frac{\nu+1}{\nu^2}(1+\frac{x^2}{\nu})^{-(\nu+7)/2}x[\nu+5-x^2\frac{(2+\nu)(5+\nu)}{\nu}+2(2+\nu)+\frac{x^2}{\nu}2(2+\nu)](\frac{-1}{\sigma^3})$$
$$=c\frac{\nu+1}{\nu^2}(1+\frac{x^2}{\nu})^{-(\nu+7)/2}x[-\frac{(2+\nu)(3+\nu)}{\nu}x^2+3(3+\nu)](\frac{-1}{\sigma^3})$$
$$=c\frac{(\nu+1)(\nu+3)}{\nu^2}(1+\frac{x^2}{\nu})^{-(\nu+7)/2}x[-\frac{2+\nu}{\nu}x^2+3](\frac{-1}{\sigma^3}).$$

The second derivative evaluations at the third derivative roots $f=y$ ($x=0$) and $f=y\pm\sigma\sqrt{\frac{3\nu}{2+\nu}}$ ($x^2=\frac{3\nu}{2+\nu}$) are $-\frac{c}{\sigma^2}\frac{\nu+1}{\nu}$ and $2\frac{c}{\sigma^2}\frac{\nu+1}{\nu}(\frac{\nu+2}{\nu+5})^{(\nu+5)/2}$, respectively. Also, the denominator of the second derivative is a polynomial of degree at least 5 whereas the numerator is a polynomial of degree 2 implying that as f approaches $\pm\infty$, the second derivative approaches 0. Hence, the second derivative is bounded over its domain.

4 Complete Experimental Details

Training: For regression, the algorithms are implemented in PyTorch. DLM is implemented as described in the main paper. Where simplified objectives are available, specifically regression ELBO for SVGP and regression objective for FITC, we implement the collapsed forms. For classification and count prediction, we extend the implementation from GPyTorch (Gardner et al., 2018). Isotropic RBF kernels are used unless otherwise specified. (We also repeated all experiments with Matern kernels and there is no big difference.) We use a zero mean function for experiments in regression and count prediction and a constant mean function for binary prediction (because some of the datasets require this to obtain reasonable performance with GP).

All algorithms are trained with the Adam optimizer where we use a learning rate of 10^{-1} for batch data training and 10^{-3} for stochastic training. The same stopping criteria consisting of either convergence or max iterations is used in all cases. Almost all runs across algorithms and datasets resulted in convergence. Convergence is defined when the difference between the minimum and maximum of the loss in the last I iterations does not exceed 10^{-4}, for $I=50$ iterations in regression, and $I=20$ iterations in classification and count prediction. For square loss DLM the optimization for m has a closed form, i.e., it is optimized in one step. If the log loss does not converge, we stop when the number of iterations exceeds 5000 for regression, and 3000 for classification and count regression. Evaluations are performed on held-out test data and 5 repetitions are used to generate error bars.

Datasets: Table 1 shows the datasets used and their characteristics. In the table, "dim" refers to the number of features and M is the number of inducing points used in our experiments. Notice that in some datasets, categorical features are converted to dummy coding, i.e., we use $L-1$ binary features to represent a feature with L categories. One category is assigned the all zero code while the other $L-1$ categories are assigned to the unit vector with the corresponding entry set to 1.

dataset	type	size	dim	M
pol[1]	regression	15000	26	100
cadata[2]	regression	20640	8	206
sarcos[3]	regression	48933	21	100
song[4]	regression	515345	90	100
banana[5]	classification	5300	2	53
thyroid[4]	classification	3772	6	37
twonorm[6]	classifcation	7400	20	74
ringnorm[7]	classification	7400	20	74
airline[8]	classification	2055733	8	200
abalone[4]	count	4177	9	41
Peds1_dir0[9]	count	4000	30	40
Peds1_dir1[9]	count	4000	30	40
Peds1_dir2[9]	count	4000	30	40

Table 1: Details of datasets

Evaluation: Each regression dataset is split into portions with relative sizes 67/8/25 for training, validation and testing. For classification and count regression, we select a number of training sizes (up to 2000) and pick 10% of all data to be the validation set. From the remaining examples we randomly choose up to 1000 samples for testing (to reduce test time for the experiments). For the larger song dataset (≈ 0.5M samples in total), we randomly choose a subset of 10000 examples for test data in order to reduce the test time in experiments. To reduce run time for DLM on large datasets we use mini-batch training with batches of 6000 samples.

For the $\approx$ 2M-size airline dataset of Hensman et al. (2015), we split a 100000 test set from the full dataset, and trained on the remaining data for 20 epochs with Adam and learning rate 10^{-3}. The number of inducing points was set to 200 and the mini-batch size was 1000. Here, we used the RBF-ARD kernel. For fixed-DLM the train/evaluation protocol is as follows: SVGP was trained with all hyperparameters and variational parameters being learned; then, DLM was initialized with the learned SVGP hyperparameters which were then fixed; the DLM variational parameters were learned from scratch.

In all cases, mean negative log likelihood (NLL) $-\log E_{q(f)}p(y|f)$ is calculated on the test set. NLL is computed exactly for regression and classification. For count regression it is calculated using quadrature. Additionally, we compute test set mean squared error (MSE) in regression, mean error in classification, and mean relative error (MRE) in count regression; the latter is defined as $\frac{|\hat{y}-y|}{\max(1,y)}$, $\hat{y} = E_{q(y)}[y] = E_{q(f)q(y|f)}[y]$. $\hat{y}$ can be calculated analytically as $E_{q(y|f)}[y] = \lambda = e^f$ and $E_{q(f)}[e^f]$ is the MGF of the normal distribution.

All datasets are normalized with respect to training data and the same normalization is performed on validation and test data.

Results: Here, we include the complete experimental results stated in the main paper.

Log-loss and sq-loss in sGP Regression and β values: Figure 1 shows results for log loss in regression. In 3 of the datasets joint-DLM is significantly better than other algorithms and in *cadata*, where hyperparameter selection is sensitive, fixed-DLM is significantly better than other algorithms. Figure 1 also shows values selected for β on small and large train sizes for the *cadata* dataset. As discussed in the main paper this illustrates that values of β larger than 1 are needed in some cases.

Figure 2 shows results for square loss on the same datasets. Note that in addition to the previous algorithms here

[1] `https://github.com/trungngv/fgp/tree/master/data/pol`
[2] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression/cadata`
[3] (Rasmussen and Williams, 2006)
[4] `http://archive.ics.uci.edu/ml/index.php`
[5] `https://www.kaggle.com/saranchandar/standard-classification-banana-dataset`
[6] `https://www.cs.toronto.edu/~delve/data/twonorm/desc.html`
[7] `https://www.cs.toronto.edu/~delve/data/ringnorm/desc.html`
[8] (Hensman et al., 2015)
[9] `http://visal.cs.cityu.edu.hk/downloads/`

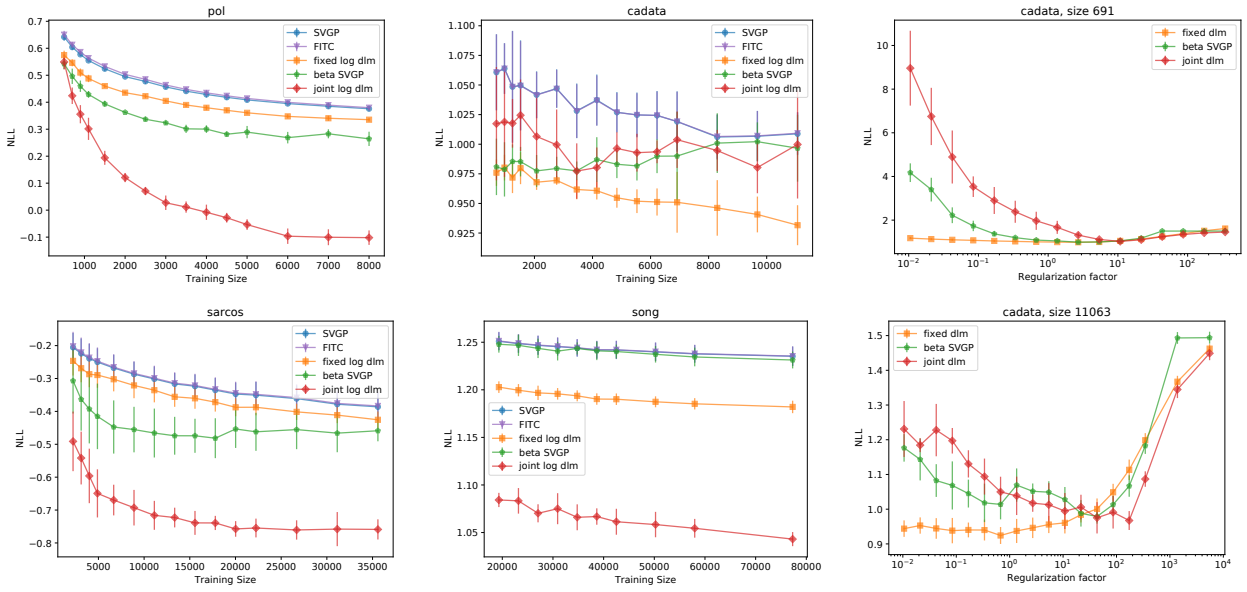

Figure 1: sGP Regression: Left and middle columns show a comparison of SVGP, FITC and DLM on mean test NLL in 4 datasets. The right column shows NLL as a function of β for cadata for a small training size and a large training size. In all plots, lower values imply better performance.

we show results for DLM that optimizes for square loss. The same pattern is repeated here where joint-sq-DLM dominates in 3 of the datasets and fixed-sq-DLM dominates in *cadata*. Note that sq-DLM algorithms improve over log-DLM algorithms for square loss.

Log-loss DLM in non-conjugate sGP: Figure 3 shows log loss in classification, and Figure 4 shows the corresponding classification error in the same experiments. In this case except for ringnorm the differences are small and DLM variants are comparable to SVGP variants.

Figure 5 shows log loss in count regression, and Figure 6 shows relative error in the same experiments. For log loss joint-DLM dominates in 3 of the datasets and fixed-DLM is equal or better in the 4th dataset. Figure 6 shows that the better calibrated prediction in terms of log loss is achieved while maintaining competitive MRE.

Non-conjugate DLM on a large dataset: Figure 7 shows a comparison between SVGP and the two DLM variants on the airline dataset for three values of β. As observed in the main paper, for this dataset, both DLM variants perform better than SVGP for all values of β tested.

Evaluation of the sampling algorithms (bias statistics): Figure 8 shows statistics of the gradients for the mean variables using uPS, bMC and smooth-bMC. The statistics for the gradients are collected immediately after the initialization of the algorithm. We show statistics for conditions similar to (i,ii) in Proposition 2 of the main paper as an estimate of the bias as compared to exact gradients. uPS is well behaved for all 3 measures. We observe that bMC with 1 sample is significantly more noisy. For the other cases the constant for condition (i) is roughly 1 (as would be with the true gradient) and the norm in condition (ii) is closer to the true gradient. The bias for bMC is significantly larger than uPS. smooth-bMC reduces the bias of bMC without negative effect on conditions (i,ii). However, as discussed below this does not lead to improvements in log loss.

Evaluation of the sampling algorithms (learning comparison):

Figures 9, 10, and 11 compare learning with exact gradients to learning with bMC and uPC for $\beta = 0.1, 1, 10$ respectively on the *airline* dataset. We observe that with enough samples both uPS and bMC recover the result of exact gradients, but uPS can do so with less samples. Figure 12 compares learning with exact gradients to learning with bMC and smooth-bMC when $\beta = 0.1$. Smooth-bMC is very close to bMC when the number of samples are the same and thus, they overlap with each other. Hence in this case the potential improvement in bias resulting from smoothing does not lead to performance improvement in terms of log loss.

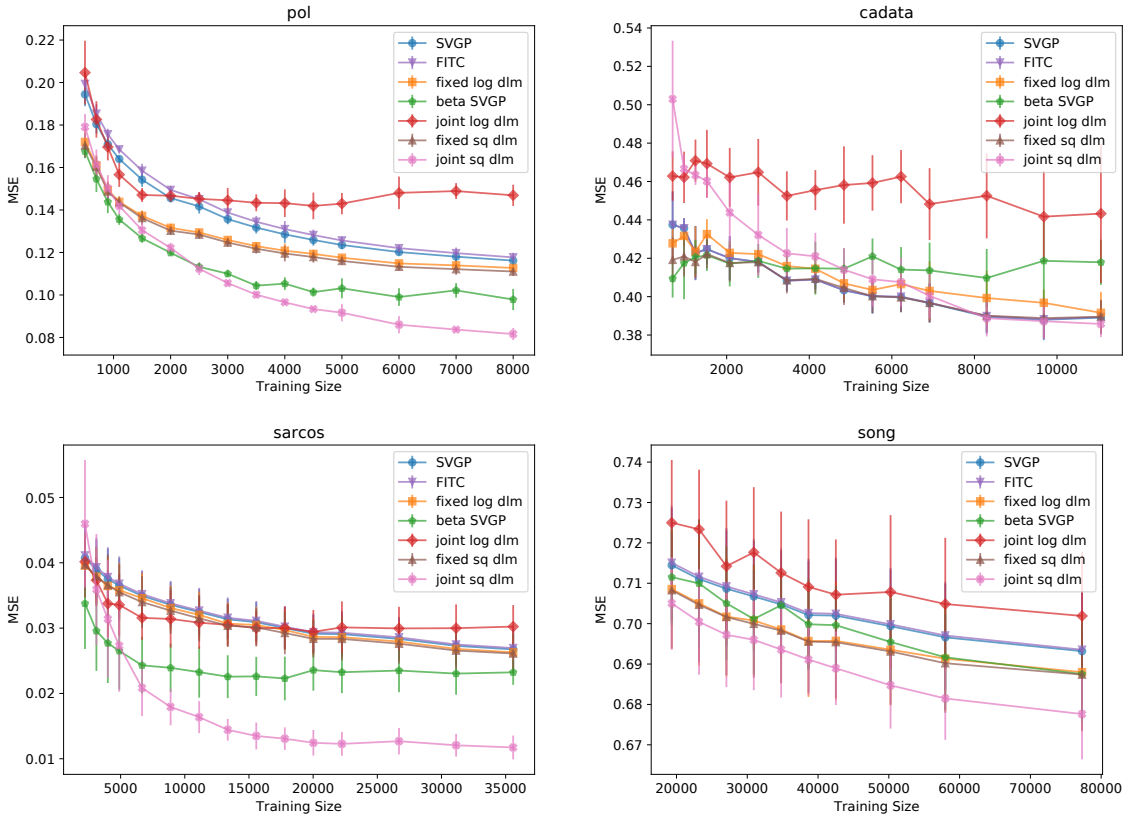

Figure 2: Square loss in sGP Regression: Comparison of SVGP, FITC, DLM and SQ_DLM in MSE. In all plots, lower values imply better performance.

Figure 13 shows learning curves for count prediction on two datasets, comparing bMC and uPS sampling. Figure 14 compares bMC and smooth-bMC. In these experiments we have found uPS to be more sensitive and have reduced the learning rate for Adam from 0.1 to 0.01. Here there are no significant differences between uPS, bMC and smooth bMC in terms of log loss.

Finally, Figure 15 compares learning with exact gradients to learning with bMC on the 4 regression datasets. For all datasets, bMC-1 is the worst and there is almost no difference between exact and bMC-100.

In summary all the experiments suggest that with enough samples bMC results in competitive performance. uPS makes better use of samples in some cases. However, this comes with a significant cost in terms or run time. Hence bMC appears to be a better choice in practice.

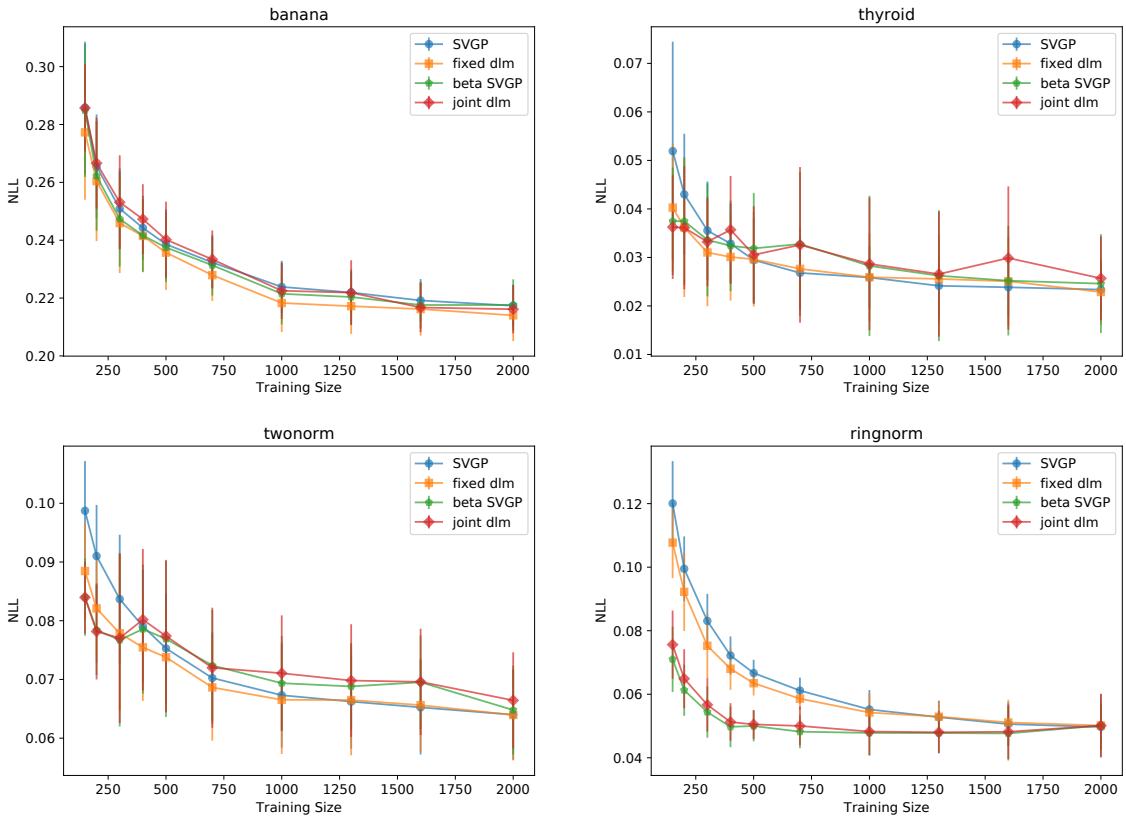

Figure 3: sGP Classification: Comparison of SVGP and DLM in mean NLL. In all plots, lower values imply better performance.

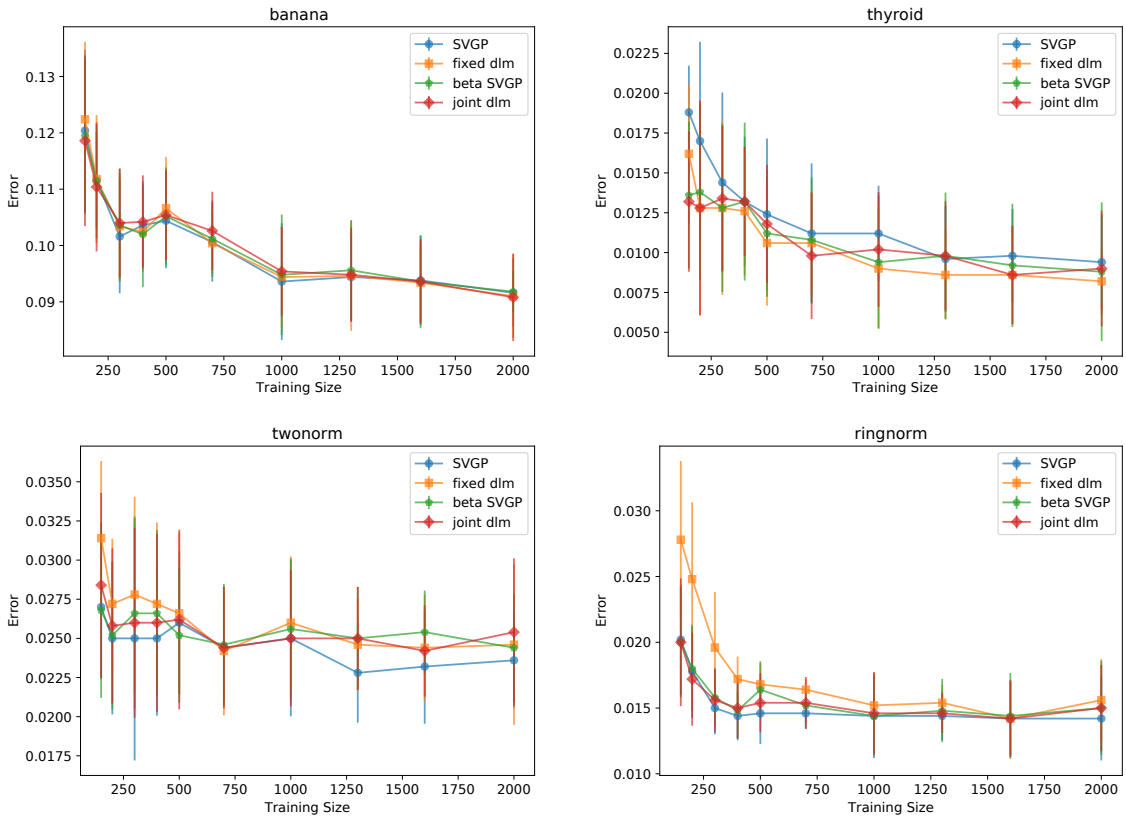

Figure 4: sGP Classification: Comparison of SVGP and DLM in term of mean error. In all plots, lower values imply better performance.

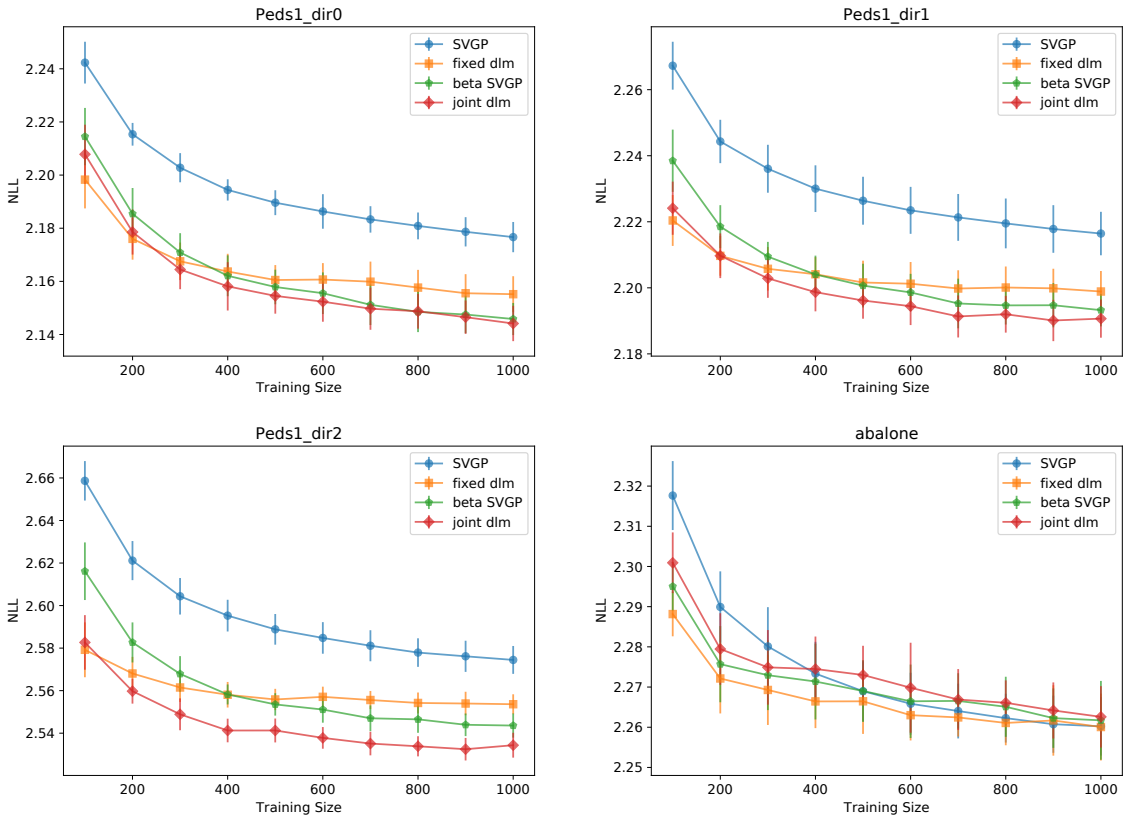

Figure 5: sGP Count Prediction: Comparison of SVGP and DLM with 10 MC samples in terms of mean NLL. In all plots, lower values imply better performance.

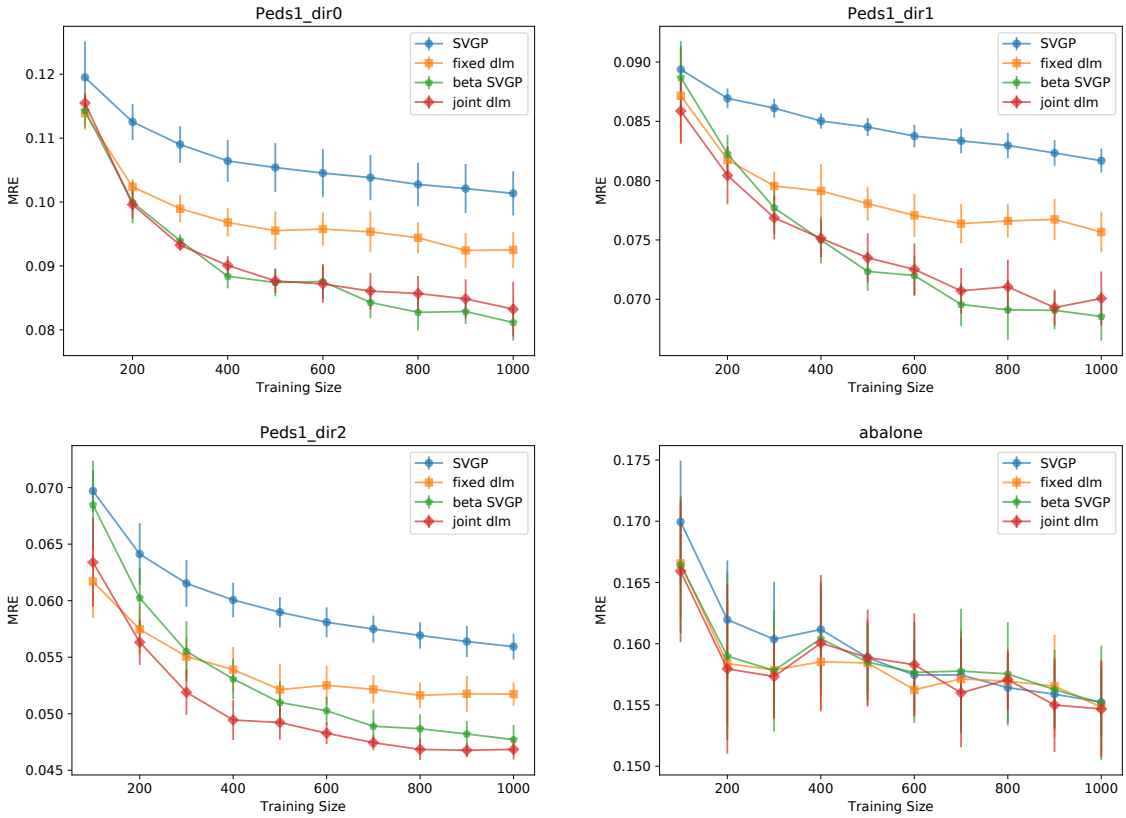

Figure 6: sGP Count Prediction: Comparison of SVGP and DLM with 10 MC samples in terms of MRE. In all plots, lower values imply better performance.

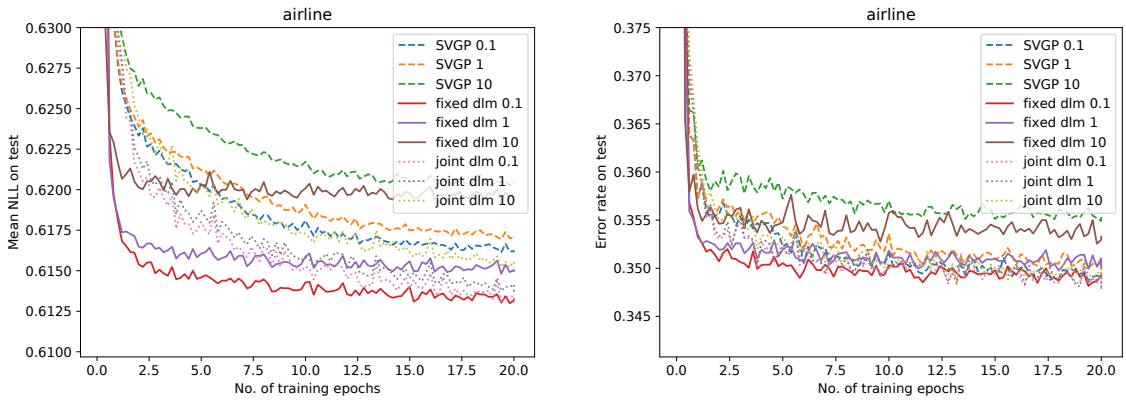

Figure 7: Comparison of SVGP and DLM with exact gradients on the binary classification airline dataset. On the left is mean NLL and on the right is mean error. In both plots, lower values imply better performance.

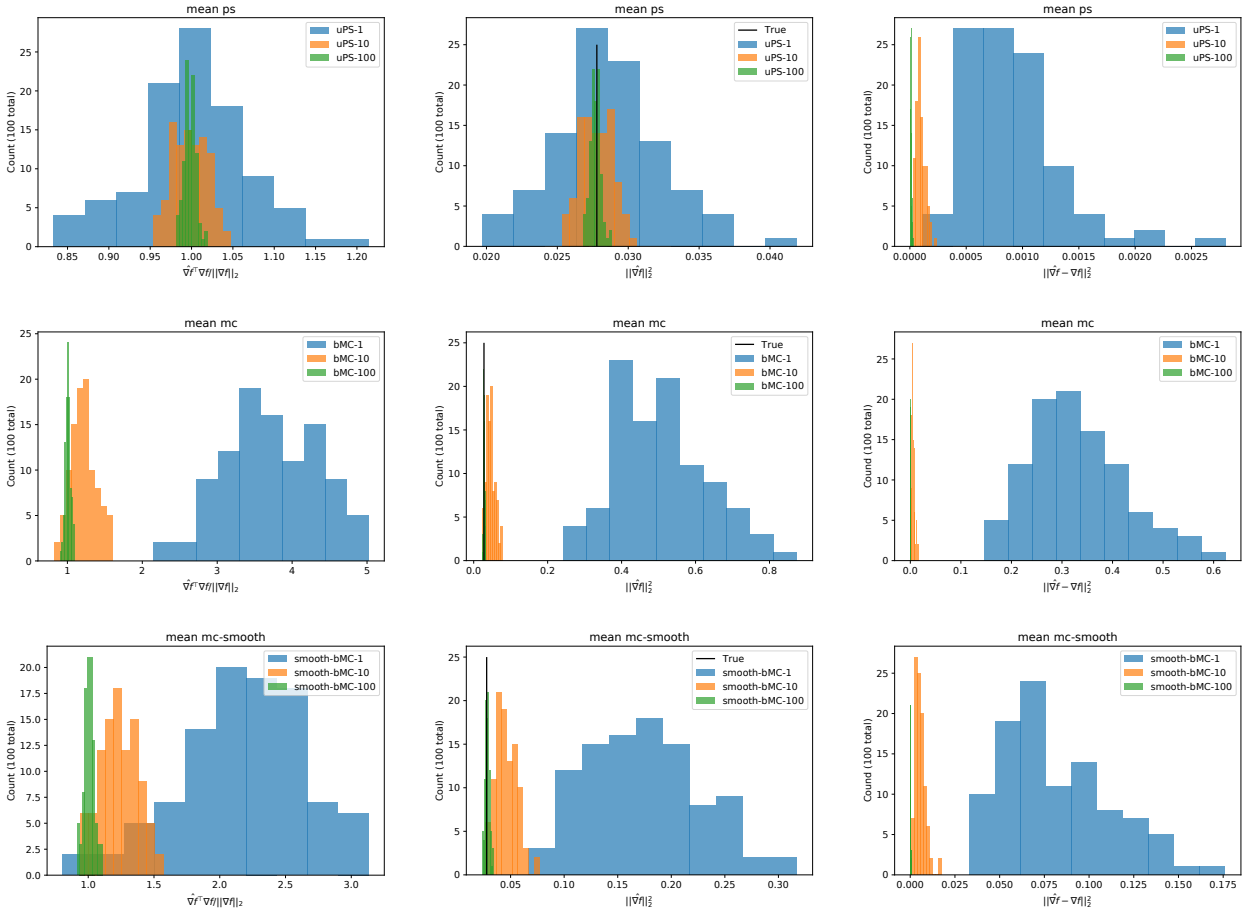

Figure 8: Statistics for calculation of biased gradients for the mean parameter for Count prediction in the Abalone dataset. First row uPS, second row bMC, and third row smooth-bMC($\nu = 10^{-4}$). Left: condition (i). Middle: condition (ii). Right: estimate of bias. Exact gradients estimated from 10000 bMC samples.

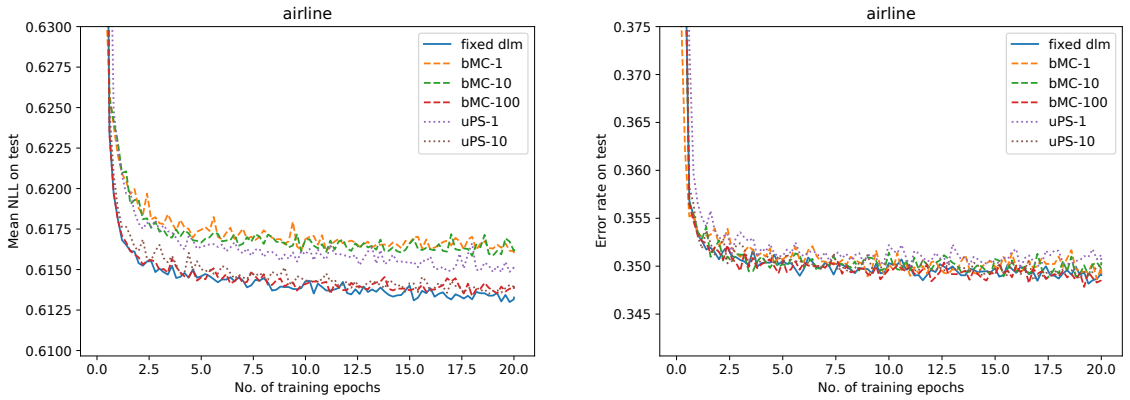

Figure 9: Comparison of DLM with exact gradients, bMC gradients and uPS gradients with $\beta = 0.1$ on the binary classification airline dataset. On the left is mean NLL and on the right is mean error. In both plots, lower values imply better performance.

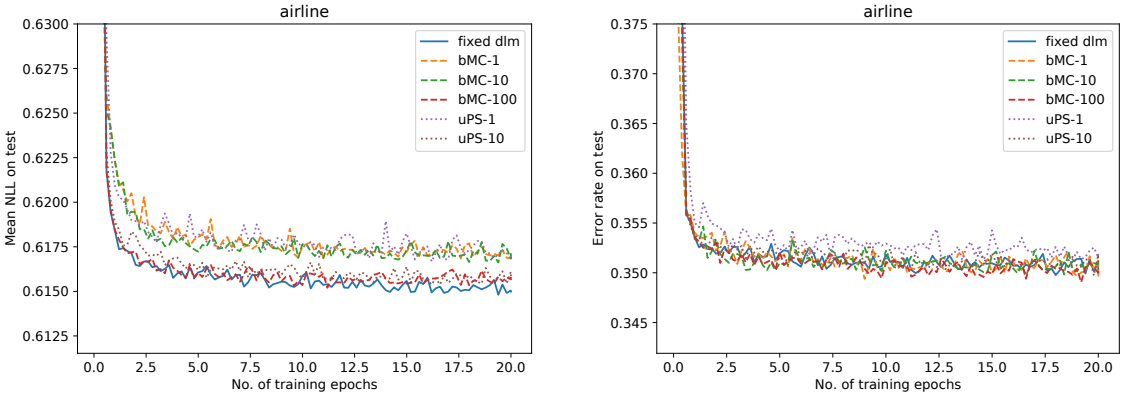

Figure 10: Comparison of DLM with exact gradients, bMC gradients and uPS gradients with $\beta = 1$ on the binary classification airline dataset. On the left is mean NLL and on the right is mean error. In both plots, lower values imply better performance.

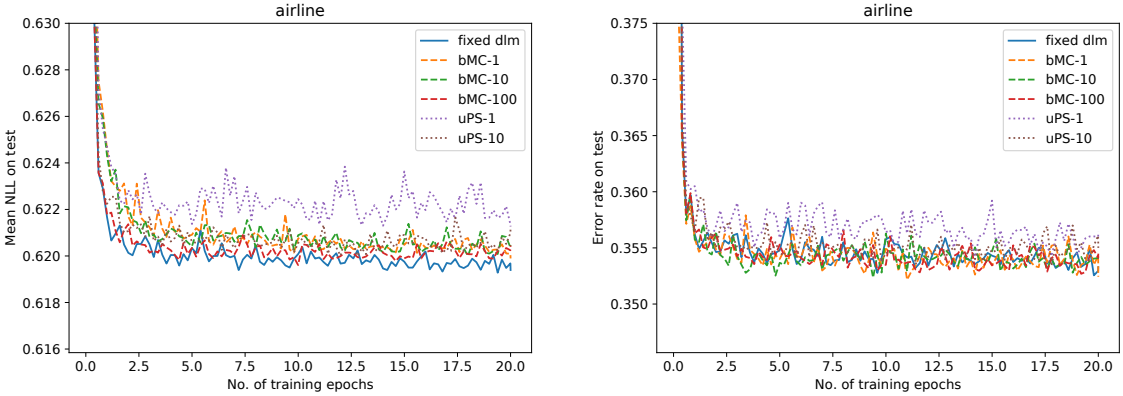

Figure 11: Comparison of DLM with exact gradients, bMC gradients and uPS gradients with $\beta = 10$ on the binary classification airline dataset. On the left is mean NLL and on the right is mean error. In both plots, lower values imply better performance.

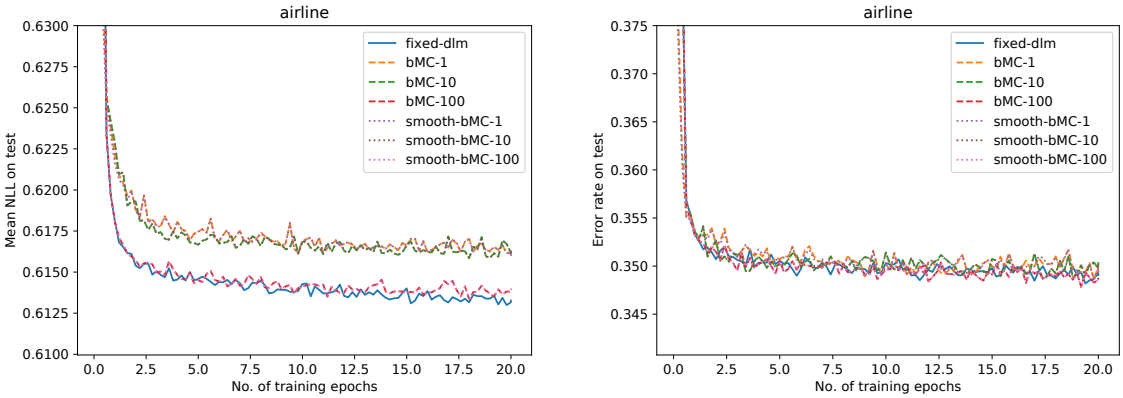

Figure 12: Learning curve of bMC and smooth-bMC, $\beta = 1$.

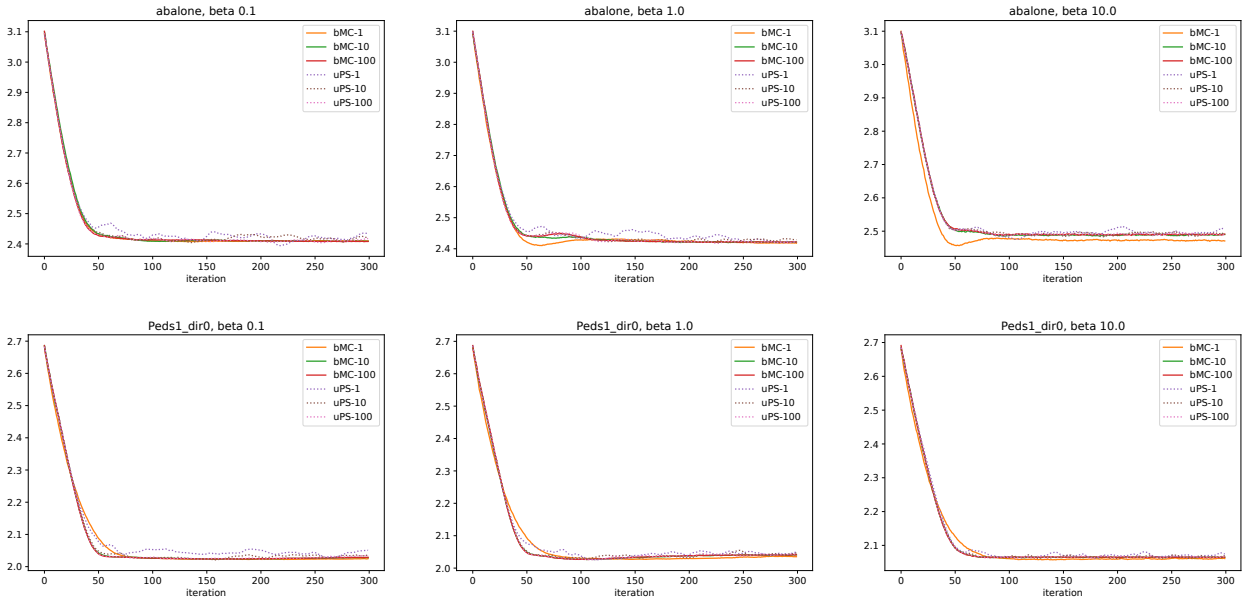

Figure 13: Comparison of uPS and bMC on two datasets for Count Prediction.

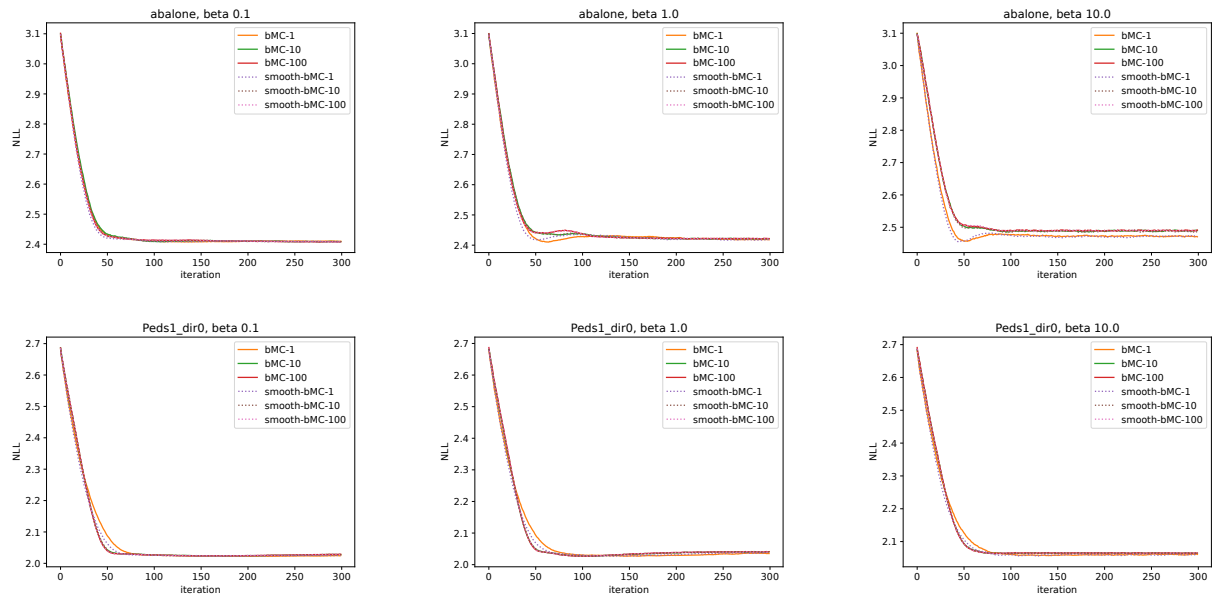

Figure 14: Comparison of bMC and smooth-bMC on two datasets for Count Prediction.

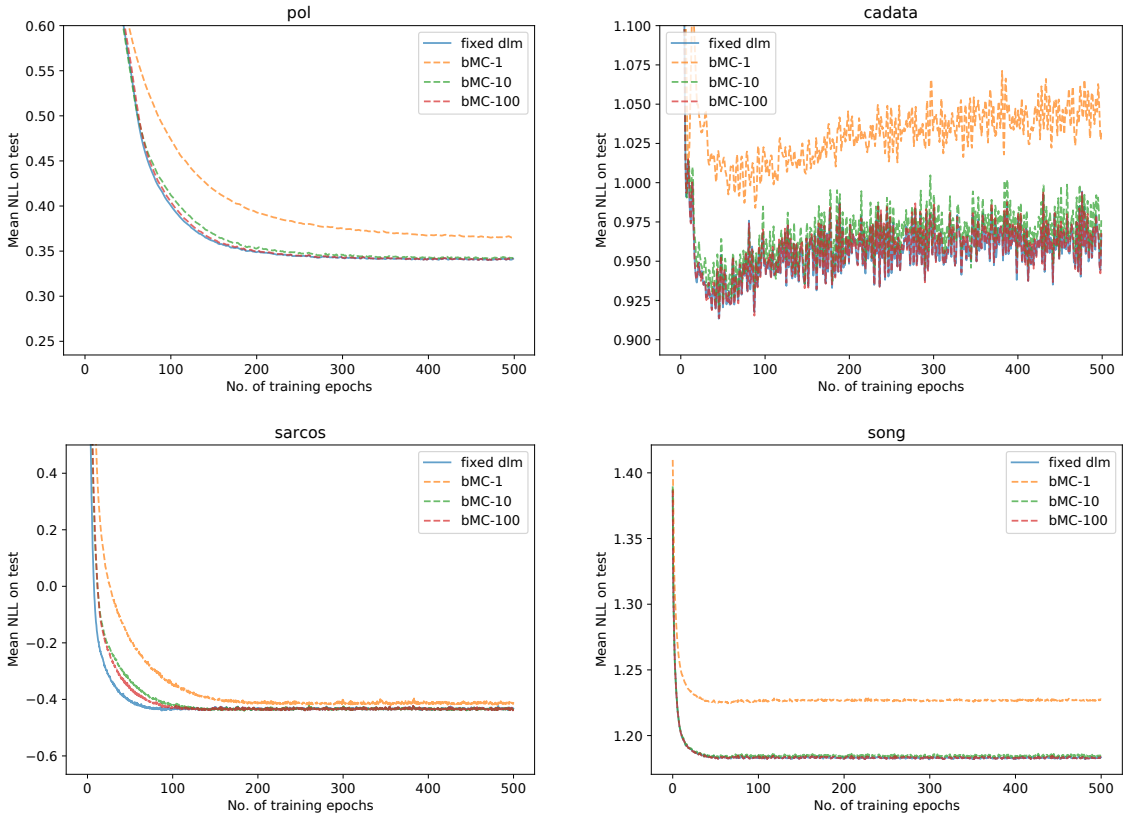

Figure 15: Comparison of exact and bMC on four datasets for regression when $\beta = 0.1$.

References

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming.* Athena Scientific.

Gardner, J. R., Pleiss, G., Bindel, D., Weinberger, K. Q., and Wilson, A. G. (2018). Gpytorch: Blackbox matrix-matrix gaussian process inference with GPU acceleration. *CoRR*, abs/1809.11165.

Hensman, J., Matthews, A., and Ghahramani, Z. (2015). Scalable variational Gaussian process classification. *JMLR.*

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