Auxiliary gradient-based sampling algorithms

Michalis K. Titsias

Athens University of Economics and Business

joint work with Omiros Papaspiliopoulos

Problem setup

We wish to sample from

$$\pi(\mathbf{x}) \propto \exp\{f(\mathbf{x})\}\pi_0(\mathbf{x})$$

- $\mathbf{x}$: latent vector of dimension n
- $\pi_0(\mathbf{x})$: tractable (Gaussian) prior
- $f(\mathbf{x})$: log-likelihood

Applications

i Gaussian process models in machine learning and statistics (e.g. multi-class Gaussian process classification)

ii Bayesian inverse problems

iii Gaussian Markov random fields

iv Deep learning can also have as smaller building blocks models of the above form, e.g. deep Gaussian processes, deep latent Gaussian models and others

Problem setup

We wish to sample from

$$\pi(\mathbf{x}) \propto \exp\{f(\mathbf{x})\}\pi_0(\mathbf{x})$$

- $\mathbf{x}$: latent vector of dimension n
- $\pi_0(\mathbf{x})$: tractable (Gaussian) prior
- $f(\mathbf{x})$: log-likelihood

Challenges for MCMC

i Likelihood-informed proposals $\Rightarrow$ effective use of the gradient $\nabla f(\mathbf{x})$

ii Prior-informed proposals $\Rightarrow$ reversibility wrt prior $\pi_0(\mathbf{x})$

iii Computational efficient samplers $\Rightarrow$ complexity should be no more than $O(n^2)$

iv Black box schemes $\Rightarrow$ work well in a large variety of problems with easy and automatic tuning

Standard gradient-based sampling

Two popular algorithms

(i) Preconditioned Metropolis-adjusted Langevin (pMALA)

$$q(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x} + \frac{\delta}{2}\mathbf{\Sigma}\nabla \log \pi(\mathbf{x}), \delta\mathbf{\Sigma})$$

and (ii) Riemann manifold MALA (Girolami and Calderhead, 2011)

$$q(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{x} + \frac{\delta}{2}\mathbf{\Sigma}(\mathbf{x})\nabla \log \pi(\mathbf{x}), \delta\mathbf{\Sigma}(\mathbf{x}))$$

Major drawbacks

- Non reversibility wrt prior $\pi_0(\mathbf{x})$
- Manifold MALA generally is computationally too expensive since it scales as $O(n^3)$
- In high-dimensional settings these algorithms become problematic

There are better schemes such as preconditioned Crank-Nicolson Langevin (pCNL); see Beskos et al (2008) and Cotter et al (2013). But the algorithms we propose here experimentally show to be 10 up to 100 times more efficient

Auxiliary gradient-based sampling

The new gradient-based samplers will make use of auxiliary variables

The starting point of the whole framework will be an auxiliary variable interpretation of the basic MALA

Auxiliary gradient-based sampling

Introduce auxiliary variables following a random walk proposal $\mathbf{u} \sim \mathcal{N}(\mathbf{u}|\mathbf{x}, (\delta/2)\mathbf{I})$ and augment the target

$$\pi(\mathbf{x}, \mathbf{u}) \propto \pi(\mathbf{x}) \times \mathcal{N}(\mathbf{u}|\mathbf{x}, (\delta/2)\mathbf{I}) = \textit{intractable}(\mathbf{x}) \times \textbf{tractable}(\mathbf{x})$$

Auxiliary MALA sampler (Metropolis-Hastings within Gibbs)

1. $\mathbf{u} \sim \mathcal{N}(\mathbf{u}|\mathbf{x}, (\delta/2)\mathbf{I})$
2. $\mathbf{y} \sim q(\mathbf{y}|\mathbf{u}, \mathbf{x})$ where the proposal is obtained by first order Taylor approximation to the intractable part:

$$\begin{aligned} q(\mathbf{y}|\mathbf{u}, \mathbf{x}) &\propto \exp\{\log \pi(\mathbf{x}) + \nabla \log \pi(\mathbf{x})^T (\mathbf{y} - \mathbf{x})\} \mathcal{N}(\mathbf{u}|\mathbf{y}, (\delta/2)\mathbf{I}) \\ &\propto \mathcal{N}(\mathbf{y}|\mathbf{u} + (\delta/2)\nabla \log \pi(\mathbf{x}), (\delta/2)\mathbf{I}) \end{aligned}$$

3. Accept $\mathbf{y}$ based on M-H step

The marginal proposal is just MALA[1]

$$q(\mathbf{y}|\mathbf{x}) = \int q(\mathbf{y}|\mathbf{u}, \mathbf{x}) \mathcal{N}(\mathbf{u}|\mathbf{x}, (\delta/2)\mathbf{I}) d\mathbf{u} = \mathcal{N}(\mathbf{y}|\mathbf{x} + (\delta/2)\nabla \log \pi(\mathbf{x}), \delta\mathbf{I})$$

[1] Due to the different acceptance mechanism the auxiliary scheme is worse in terms of sampling efficiency (Peskun ordering) than its marginal MALA version

Auxiliary gradient-based sampling

$$\pi(\mathbf{x}, \mathbf{u}) \propto \pi(\mathbf{x}) \times \mathcal{N}(\mathbf{u}|\mathbf{x}, (\delta/2)\mathbf{I}) = \textit{intractable}(\mathbf{x}) \times \textit{tractable}(\mathbf{x})$$

Auxiliary MALA sampler

1. $\mathbf{u} \sim \mathcal{N}(\mathbf{u}|\mathbf{x}, (\delta/2)\mathbf{I})$
2. $\mathbf{y} \sim q(\mathbf{y}|\mathbf{u}, \mathbf{x})$ where

$$q(\mathbf{y}|\mathbf{u}, \mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{u} + (\delta/2)\nabla \log \pi(\mathbf{x}), (\delta/2)\mathbf{I})$$

3. Accept $\mathbf{y}$ based on M-H step

This auxiliary variable interpretation of MALA give us a lot of freedom to derive many novel algorithms that are very different than MALA

How?

- **by just moving tractable parts from** $\textit{intractable}(\mathbf{x})$ **into** $\textit{tractable}(\mathbf{x})$

Auxiliary gradient-based sampling

$$\begin{aligned}\pi(\mathbf{x},\mathbf{u}) &\propto \pi(\mathbf{x}) \times \mathcal{N}(\mathbf{u}|\mathbf{x},(\delta/2)\mathbf{I}) \\ &\propto \exp\{f(\mathbf{x})\}\pi_0(\mathbf{x}) \times \mathcal{N}(\mathbf{u}|\mathbf{x},(\delta/2)\mathbf{I}) \\ &\propto \exp\{f(\mathbf{x})\} \times \pi_0(\mathbf{x})\mathcal{N}(\mathbf{u}|\mathbf{x},(\delta/2)\mathbf{I})\end{aligned}$$

where we moved the tractable (Gaussian) $\pi_0(\mathbf{x})$ into the tractable part

Key idea: We only need to approximate the intractable log-likelihood $f(\mathbf{x})$ and not the tractable prior $\pi_0(\mathbf{x})$!

A novel auxiliary gradient-based sampler

1. $\mathbf{u} \sim \mathcal{N}(\mathbf{u}|\mathbf{x},(\delta/2)\mathbf{I})$
2. $\mathbf{y} \sim q(\mathbf{y}|\mathbf{u},\mathbf{x})$ where

$$q(\mathbf{y}|\mathbf{u},\mathbf{x}) \propto \exp\{f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x})\}\pi_0(\mathbf{y})\mathcal{N}(\mathbf{u}|\mathbf{y},(\delta/2)\mathbf{I})$$

3. Accept $\mathbf{y}$ based on M-H step

Auxiliary gradient-based sampling

A novel auxiliary gradient-based sampler

1. $\mathbf{u} \sim \mathcal{N}(\mathbf{u}|\mathbf{x}, (\delta/2)\mathbf{I})$
2. $\mathbf{y} \sim q(\mathbf{y}|\mathbf{u}, \mathbf{x})$ where

$$q(\mathbf{y}|\mathbf{u}, \mathbf{x}) \propto \exp\{f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x})\}\pi_0(\mathbf{y})\mathcal{N}(\mathbf{u}|\mathbf{y}, (\delta/2)\mathbf{I})$$

3. Accept $\mathbf{y}$ based on M-H step

By marginalising out $\mathbf{u}$ we can obtain the corresponding marginal sampler

A novel marginal gradient-based sampler

1. $\mathbf{y} \sim q(\mathbf{y}|\mathbf{x})$ where

$$q(\mathbf{y}|\mathbf{x}) = \int q(\mathbf{y}|\mathbf{u}, \mathbf{x})\mathcal{N}(\mathbf{u}|\mathbf{x}, (\delta/2)\mathbf{I})d\mathbf{u}$$

2. Accept $\mathbf{y}$ based on M-H step

Auxiliary gradient-based sampling

These new samplers have all desired properties. I.e. they are

- Likelihood informed $\Rightarrow$ we make effective use of $\nabla f(\mathbf{x})$
- Prior-informed $\Rightarrow$ reversibility wrt prior $\pi_0(\mathbf{x})$
- Computationally efficient as we will see next

Application to latent Gaussian models

From now on let us focus on the following specific latent Gaussian model

$$\pi(\mathbf{x}) \propto \exp\{f(\mathbf{x})\}\mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{C})$$

- where $\pi_0(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{C})$ is the Gaussian prior
- we have assumed zero-mean for simplicity
- and parametrization based on the covariance matrix $\mathbf{C} \Rightarrow$ typical for Gaussian process models in machine learning
- other cases are similar

Application to latent Gaussian models

Expanded target

$$\pi(\mathbf{x}, \mathbf{u}) \propto \exp\{f(\mathbf{x})\}\mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{C})\mathcal{N}(\mathbf{u}|\mathbf{x}, (\delta/2)\mathbf{I})$$

Auxiliary sampler based on $\mathbf{u}$

1. $\mathbf{u} \sim \mathcal{N}(\mathbf{u}|\mathbf{x}, (\delta/2)\mathbf{I})$
2. Propose $\mathbf{y} \sim q(\mathbf{y}|\mathbf{x}, \mathbf{u})$ where

$$\begin{aligned} q(\mathbf{y}|\mathbf{x}, \mathbf{u}) &\propto \exp\{f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x})\}\mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{C})\mathcal{N}(\mathbf{u}|\mathbf{y}, (\delta/2)\mathbf{I}) \\ &\propto \mathcal{N}\left(\mathbf{y}|\frac{2}{\delta}\mathbf{A}(\mathbf{u} + \frac{\delta}{2}\nabla f(\mathbf{x})), \mathbf{A}\right) \end{aligned}$$

3. Accept according to the M-H ratio

$$\exp\left\{f(\mathbf{y}) - f(\mathbf{x}) + j(\mathbf{x}, \mathbf{y}, \mathbf{u}) - j(\mathbf{y}, \mathbf{x}, \mathbf{u})\right\},$$

$$j(\mathbf{x}, \mathbf{y}, \mathbf{u}) = \left(\mathbf{x} - \frac{2}{\delta}\mathbf{A}(\mathbf{u} + \frac{\delta}{4}\nabla f(\mathbf{y}))\right)^T \nabla f(\mathbf{y})$$

where $\mathbf{A} = (\mathbf{C}^{-1} + \frac{2}{\delta}\mathbf{I})^{-1} = \frac{\delta}{2}(\mathbf{C} + \frac{\delta}{2}\mathbf{I})^{-1}\mathbf{C}$

Application to latent Gaussian models

The corresponding marginal scheme is obtained by integrating out $\mathbf{u}$ to obtain the marginal proposal

Marginal sampler

1. Propose $\mathbf{y} \sim q(\mathbf{y}|\mathbf{x})$ where

$$q(\mathbf{y}|\mathbf{x}) = \int \mathcal{N}(\mathbf{y}|\frac{2}{\delta}\mathbf{A}(\mathbf{u} + (\delta/2)\nabla f(\mathbf{x})), \mathbf{A})\mathcal{N}(\mathbf{u}|\mathbf{x}, (\delta/2)\mathbf{I})\, d\mathbf{u}$$
$$= \mathcal{N}\left(\mathbf{y}|\frac{2}{\delta}\mathbf{A}\left(\mathbf{x} + \frac{\delta}{2}\nabla f(\mathbf{x})\right), \frac{2}{\delta}\mathbf{A}^2 + \mathbf{A}\right),$$

2. Accept according to the M-H ratio

$$\exp\{f(\mathbf{y}) - f(\mathbf{x}) + h(\mathbf{x}, \mathbf{y}) - h(\mathbf{y}, \mathbf{x})\},$$
$$h(\mathbf{x}, \mathbf{y}) = \left(\mathbf{x} - \frac{2}{\delta}\mathbf{A}(\mathbf{y} + \frac{\delta}{4}\nabla f(\mathbf{y}))\right)^T \left(\frac{2}{\delta}\mathbf{A} + \mathbf{I}\right)^{-1} \nabla f(\mathbf{y})$$

Application to latent Gaussian models

Reparameterize $\mathbf{z} \equiv \mathbf{u} + (\delta/2)\nabla f(\mathbf{x})$ and work with an alternative expanded target

$$\pi(\mathbf{x}, \mathbf{z}) \propto \exp\{f(\mathbf{x})\}\mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{C})\mathcal{N}(\mathbf{z}|\mathbf{x} + (\delta/2)\nabla f(\mathbf{x}), (\delta/2)\mathbf{I})$$

Auxiliary sampler based on $\mathbf{z}$

1. $\mathbf{z} \sim \mathcal{N}(\mathbf{z}|\mathbf{x} + (\delta/2)\nabla f(\mathbf{x}), (\delta/2)\mathbf{I})$
2. Propose $\mathbf{y} \sim q(\mathbf{y}|\mathbf{z})$ where

$$q(\mathbf{y}|\mathbf{z}) = \mathcal{N}(\mathbf{y}|(2/\delta)\mathbf{A}\mathbf{z}, \mathbf{A}) \propto \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{C})\mathcal{N}(\mathbf{z}|\mathbf{y}, (\delta/2)\mathbf{I})$$

3. Accept according to M-H ratio

$$\exp\{f(\mathbf{y}) - f(\mathbf{x}) + g(\mathbf{z}, \mathbf{y}) - g(\mathbf{z}, \mathbf{x})\},$$
$$g(\mathbf{z}, \mathbf{y}) = (\mathbf{z} - \mathbf{y} - (\delta/4)\nabla f(\mathbf{y}))^T \nabla f(\mathbf{y})$$

The M-H step scales as $O(n)$ $\Rightarrow$ this scheme allows a tradeoff between computational and statistical efficiency

Application to latent Gaussian models

Auxiliary sampler based on $\mathbf{z}$

1. $\mathbf{z} \sim \mathcal{N}(\mathbf{z}|\mathbf{x} + (\delta/2)\nabla f(\mathbf{x}), (\delta/2)\mathbf{I})$
2. Propose $\mathbf{y} \sim q(\mathbf{y}|\mathbf{z})$ where

$$q(\mathbf{y}|\mathbf{z}) = \mathcal{N}(\mathbf{y}|(2/\delta)\mathbf{A}\mathbf{z}, \mathbf{A}) \propto \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{C})\mathcal{N}(\mathbf{z}|\mathbf{y}, (\delta/2)\mathbf{I})$$

3. Accept according to M-H ratio

$$\exp\{f(\mathbf{y}) - f(\mathbf{x}) + g(\mathbf{z}, \mathbf{y}) - g(\mathbf{z}, \mathbf{x})\},$$
$$g(\mathbf{z}, \mathbf{y}) = (\mathbf{z} - \mathbf{y} - (\delta/4)\nabla f(\mathbf{y}))^T \nabla f(\mathbf{y})$$

This schemes has many interesting computational properties

- $\mathbf{z}$ renders the proposed $\mathbf{y}$ conditionally independent from the current $\mathbf{x}$
- step 1 is only likelihood-informed while step 2 is only prior-informed
- these two types of information are linked through $\mathbf{z}$

Such scheme has relatively long history, derived in Titsias (2011)

Application to latent Gaussian models

Given a precomputed eigendecomposition of the prior covariance $\mathbf{C}$

- All algorithms require overall $O(n^2)$ time per a full iteration
- This is because all matrices involved such as $\mathbf{A} = (\mathbf{C}^{-1} + \frac{2}{\delta}\mathbf{I})^{-1}$ share the same eigenvectors with $\mathbf{C}$
- The step size δ is tuned to achieve an acceptance rate between $50\% - 60\%$ which empirically we have observed to maximize sampling efficiency

Experiments

We compare several schemes

- the marginal sampler (mGrad)
- the sampler based on auxiliary variable $\mathbf{u}$ (aGrad-u)
- the sampler based on $\mathbf{z}$ (aGrad-z)
- preconditioned MALA (pMALA) with proposal

$$q(\mathbf{y}|\mathbf{x}) = \mathcal{N}\left(\mathbf{y}|(1-\frac{\delta}{2})\mathbf{x} + \frac{\delta}{2}\mathbf{C}\nabla f(\mathbf{x}), \delta\mathbf{C}\right)$$

- preconditioned Crank-Nicolson (pCN)

$$q(\mathbf{y}|\mathbf{x}) = \mathcal{N}\left(\mathbf{y}|\frac{2}{2+\delta}\mathbf{x}, \frac{\delta(\delta+4)}{(2+\delta)^2}\mathbf{C}\right)$$

- preconditioned Crank-Nicolson Langevin (pCNL)

$$q(\mathbf{y}|\mathbf{x}) = \mathcal{N}\left(\mathbf{y}|\frac{2}{2+\delta}\mathbf{x} + \frac{\delta}{2+\delta}\mathbf{C}\nabla f(\mathbf{x}), \frac{\delta(\delta+4)}{(2+\delta)^2}\mathbf{C}\right)$$

- elliptical slice sampling (Ellipt) of Murray and Adams (2010), which is very popular in the machine learning community

Experiments

Gaussian process regression

$$f(\mathbf{x}) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - x_i)^2$$

Experiments

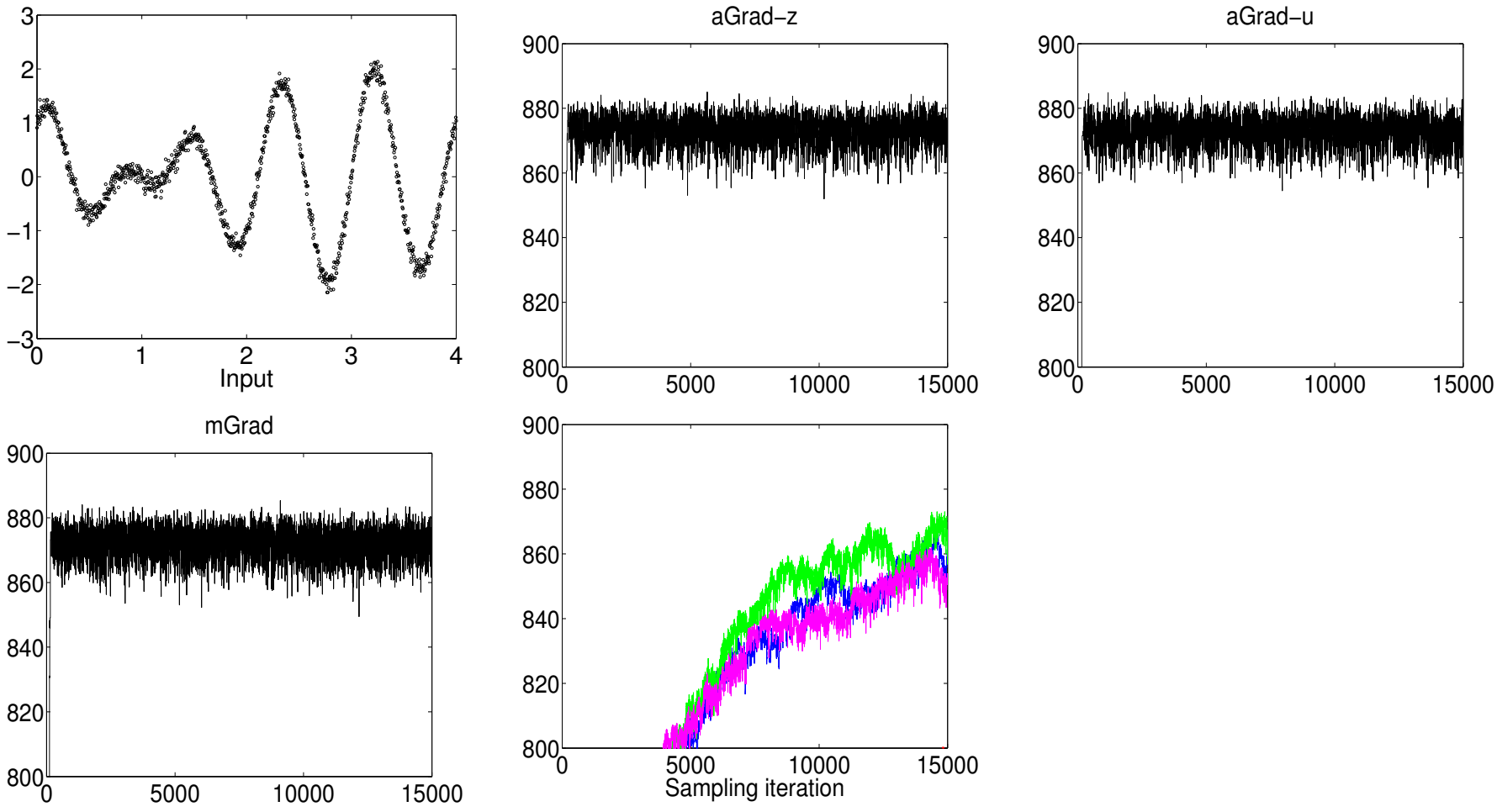

Figure: First panel shows the dataset. The next panels show the evolution of $f(\mathbf{x})$ across iterations for all algorithms.

Experiments

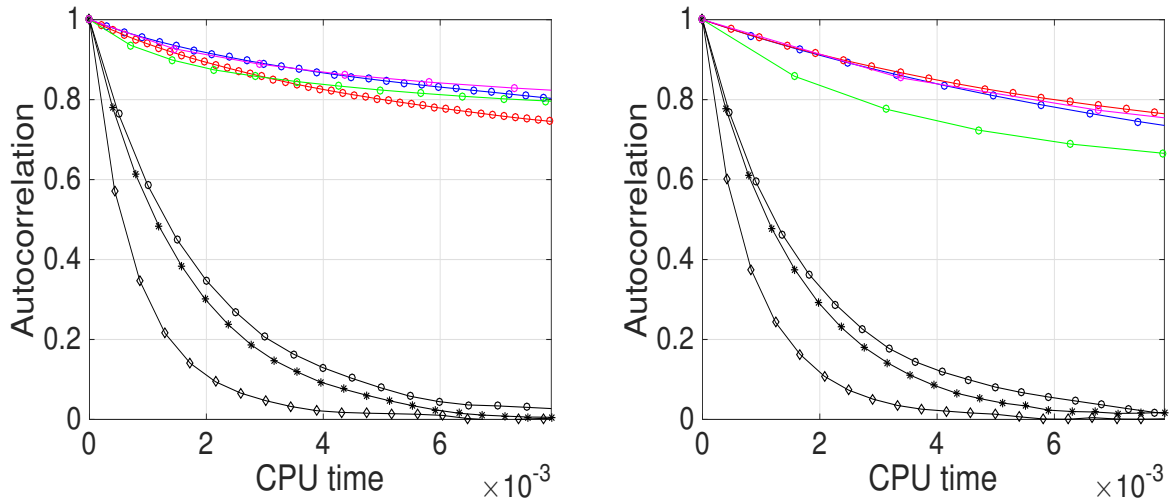

Figure: Estimated autocorrelation of the function $f(\mathbf{x})$ for all algorithms against CPU time; the lines are averages over ten runs of the algorithms.

Experiments

Table: Comparison of sampling methods in the regression dataset with $\sigma^2 = 0.01$. All numbers are averages across ten repeats where also one-standard deviation is given for the Min ESS/s score.

Method	*Time(s)*	*Step* δ	*ESS (Min, Med, Max)*	*Min ESS/s (s.d.)*
aGrad-z	5.5	0.005	(306.7, 430.9, 539.3)	55.48 (5.96)
aGrad-u	6.9	0.006	(345.8, 459.3, 584.8)	50.27 (4.02)
mGrad	5.8	0.011	(856.0, 1095.6, 1301.1)	147.67 (9.97)
pMALA	49.3	< 0.001	(8.4, 45.1, 155.5)	0.17 (0.07)
Ellipt	11.0		(12.9, 50.3, 144.8)	1.18 (0.34)
pCN	6.6	< 0.001	(8.0, 35.4, 107.7)	1.21 (0.42)
pCNL	23.9	< 0.001	(8.7, 54.4, 208.6)	0.36 (0.13)

- The new samplers can be around 100 better than the other schemes
- Notice that the adapted step size δ for the new samplers is close to the noise σ^2 of the likelihood

Experiments

Log-Gaussian Cox Process

$$f(\mathbf{x}) = \sum_{i,j}^{64} (y_{ij}(x_{ij} + v) - m \exp(x_{ij} + v))$$

The consider the same dataset used by Girolami and Calderhead (2011)

The dimensionality of $\mathbf{x}$ is $n = 4096$

In this example we also compare with manifold MALA (mMALA) and the Riemann Manifold Hamiltonian Monte Carlo (RMHMC) using the code of Girolami and Calderhead

Experiments

Table: Comparison of sampling methods in the log-Gaussian Cox model dataset in the original version where $n = 4096$. All numbers are averages across ten repeats where also one-standard deviation is given for the Min ESS/s score.

Method	*Time(s)*	*Step* δ	*ESS (Min, Med, Max)*	*Min ESS/s (s.d.)*
aGrad-z	89.6	0.962	(36.1, 181.2, 507.5)	0.40 (0.10)
aGrad-u	134.6	2.814	(95.8, 469.3, 1092.4)	0.71 (0.14)
mGrad	132.0	5.887	(177.8, 801.1, 1628.6)	1.35 (0.15)
pMALA	218.7	0.006	(3.5, 12.9, 59.3)	0.02 (0.00)
Ellipt	51.5		(4.2, 17.0, 66.0)	0.08 (0.01)
pCN	47.4	0.012	(3.3, 12.0, 57.6)	0.07 (0.00)
pCNL	87.7	0.006	(3.4, 12.6, 60.2)	0.04 (0.00)
mMALA	334.1	0.070	(21.4, 84.7, 179.4)	0.06 (0.02)
RMHMC	1493.7	0.100	(1825.7, 4452.2, 5000.0)	1.22 (0.09)

- The new samplers are around 25 times better than other schemes but RMHMC which has slightly less Min ESS/s than mGrad
- However, notice that for this log-Gaussian Cox model, and due to the analytic properties of the log-likelihood, RMHMC and mMALA use a constant metric tensor (which is not the case for more complex models)

Experiments

Multi-class Gaussian process classification on 1000 MNIST digits belonging to $K = 10$ classes with log-likelihood

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_K) = \sum_{i=1}^{n} \left(x_{y_i i} - \log \sum_{k=1}^{K} \exp\{x_{ki}\} \right)$$

where each $\mathbf{x}_k \sim \mathcal{N}(\mathbf{x}_k | \mathbf{0}, \mathbf{C}_{\boldsymbol{\theta}_k})$ and $\boldsymbol{\theta}_k$ denotes the kernel hyperparameters.

we sample jointly all 10^4 latent variables $(\mathbf{x}_1, \ldots, \mathbf{x}_K)$

we also sample kernel hyperparameters $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$

Experiments

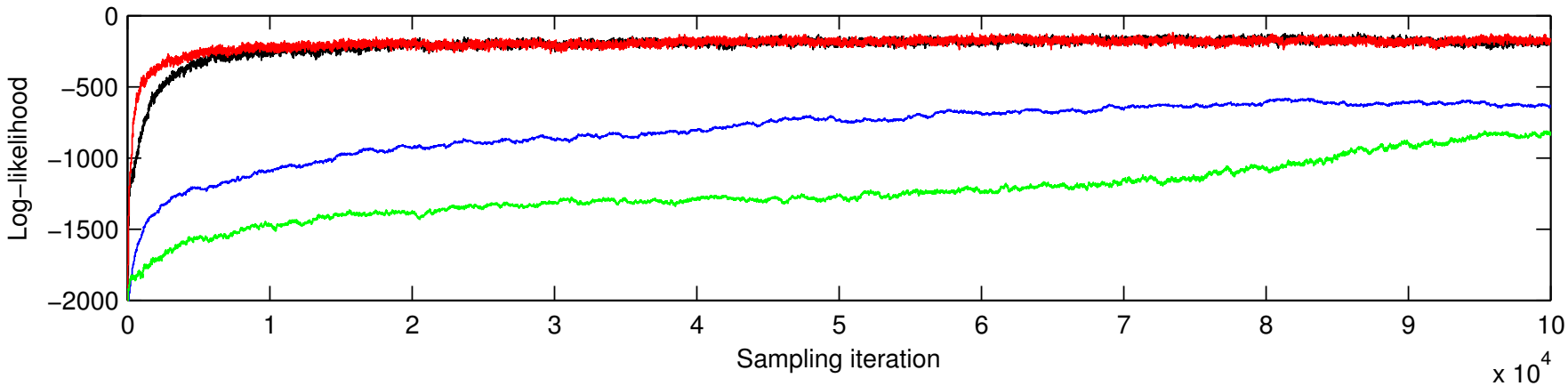

Figure: The evolution of the log-likelihood values across all iterations in MNIST dataset for the sampling schemes aGrad-z-gibbs (black line), aGrad-z-joint (red line), Ellipt-gibbs (blue line) and pCNL-gibbs (green line).

Discussion

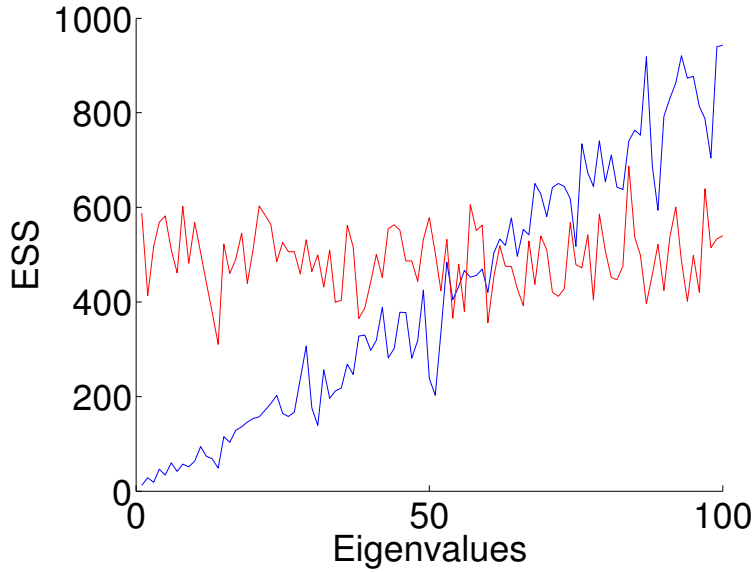

Figure: With red is effective sample size for the marginal sampler (across all components of $\mathbf{x}$) and with blue the corresponding of PCNL in a toy example with varying eigenvalues in $\mathbf{C}$.

Why pCNL with proposal $q(\mathbf{y}|\mathbf{x}) = \mathcal{N}\left(\mathbf{y}|\frac{2}{2+\delta}\mathbf{x} + \frac{\delta}{2+\delta}\mathbf{C}\nabla f(\mathbf{x}), \frac{\delta(\delta+4)}{(2+\delta)^2}\mathbf{C}\right)$ (despite of being both likelihood and prior informed) does not work well

- Because $\frac{\delta(\delta+4)}{(2+\delta)^2}\mathbf{C}$ tries to approximate the posterior covariance (of the form $(\mathbf{C}^{-1} + \Lambda)^{-1}$) by uniformly shrinking all eigenvalues of the prior covariance to fit the shape of the posterior covariance
- Thus the smaller eigenvalues will be unnecessarily shrunk too much and the associated components of $\mathbf{x}$ will move very slowly

Discussion

Summary

- We introduced a new framework for gradient-based samplers using auxiliary variables and simple Taylor approximations
- This framework gives several new algorithms that outperform by a large margin other schemes such pCN, pCNL and elliptical slice sampling
- The framework is also very general. E.g. pCNL (which is rigorously derived as SDE discretization) can be shown to be a special case

Future work

- There are many open theoretical questions
- New applications

References

- The full framework of auxiliary gradient-based samplers can be found in:

  M. K. Titsias and O. Papaspiliopoulos. Auxiliary gradient-based sampling algorithms, Journal of the Royal Statistical Society: Series B, 2018, to appear

- Earlier work that introduced the auxiliary sampler based on $\mathbf{z}$:

  Michalis K. Titsias. Contribution to the discussion of the paper by Girolami and Calderhead. Journal of the Royal Statistical Society: Series B, 73(2):197?199, 2011

- An application in sampling millions of latent variables in multivariate stochastic volatility applications:

  P. Dellaportas, A. Plataniotis, M. K. Titsias. Scalable inference for a full multivariate stochastic volatility model, arXiv:1510.05257, 2015