SAMPLING POSTERIORS IN HIGH DIMENSION

POTENTIAL INDUSTRIAL APPLICATIONS WITH UQ

—

GDR Mascot-Num Workshop March 2020

Sébastien Da Veiga
Safran Tech

Objective of the talk

Introduce some recent sampling techniques used in ML/DL

Illustrate and discuss their potential for our daily UQ applications

Outlook

Langevin Dynamics and variants for sampling

(Stein) kernels for subsampling

1

LANGEVIN DYNAMICS AND VARIANTS

Langevin Dynamics

Context: we want to sample from an unnormalized density defined in terms of a potential

$$\pi(\boldsymbol{\theta}) \propto \exp\{-U(\boldsymbol{\theta})\}$$

Under mild conditions, this density is the unique invariant probability measure of the Langevin SDE

$$\mathrm{d}\theta_t = -\nabla U(\theta_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

Langevin Dynamics – ULA

In general we cannot solve this SDE exactly: we thus rely on an approximation

- Euler (-Maruyama) method for discretization

$$\theta_{k+1} = \theta_k - \gamma \nabla U(\theta_k) + \sqrt{2\gamma} Z_{k+1}$$

This is the Unadjusted Langevin Algorihtm (ULA) or Langevin Monte Carlo (LMC)

- Simply a MAP descent algorithm with noise added at each iteration
- Recent theoretical results to control the approximation error with respect to sample size and dimension (if target is regular)
  Durmus & Moulines 2017
- **Several success for high-dimensional sampling problems in Bayesian inference**

Langevin Dynamics – MALA

Discretization induces bias, which can be removed by an additional Metropolis-Hastings accept-reject step

- **This is the Metropolis-Adjusted Langevin Algorithm (MALA)**
- **The proposals based on ULA have a much higher acceptance rate than standard random walk MH**

MALA inherits good convergence properties of ULA and scales efficiently to high-dimensional settings (Durmus et al. 2018)

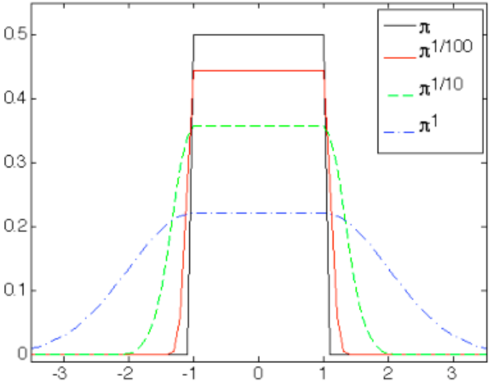
- Valid only if target is regular, again
- In practice (Nemeth & Fearnhead 2019):
  - The optimal step size for MALA is large, but MALA has higher cost per iteration
  - ULA usually requires smaller step sizes (bias) so more iterations, but has smaller cost per iteration

Langevin Dynamics – MYULA (1/2) Durmus et al. 2018

When target distribution is not regular, assume the potential can be written

$$U(\theta) = f(\theta) + g(\theta)$$

- f is convex, continuously derivable and gradient Lipschitz
- g is proper, convex and lower semi-continuous



Replage g by its Moreau-Yosida enveloppe

$$\mathrm{g}^{\lambda}(x) = \min_{y \in \mathbb{R}^d} \left\{ \mathrm{g}(y) + (2\lambda)^{-1} \|x - y\|^2 \right\}$$

$$\nabla \mathrm{g}^{\lambda}(x) = \lambda^{-1} \left(x - \mathrm{prox}_{\mathrm{g}}^{\lambda}(x) \right)$$

$$\mathrm{prox}_{\mathrm{g}}^{\lambda}(x) = \arg\min_{y \in \mathbb{R}^d} \left\{ \mathrm{g}(y) + (2\lambda)^{-1} \|x - y\|^2 \right\}$$

Langevin Dynamics – MYULA (2/2) Durmus et al. 2018

Moreau-Yosida ULA (MYULA) is then given by

$$\theta_{k+1} = \left(1 - \frac{\gamma}{\lambda}\right)\theta_k - \gamma\nabla f(\theta_k) + \frac{\gamma}{\lambda}\mathrm{prox}_g^{\lambda}(\theta_k) + \sqrt{2\gamma}Z_{k+1}$$

Langevin Dynamics – Bayesian inference

Common situation: unnormalized target distribution writes

$$\pi(\theta) = \pi_0(\theta) \prod_{i=1}^{N} p(z_i|\theta)$$

$$U = \sum_{i=0}^{N} U_i$$

$$U_0(\theta) = -\log(\pi_0(\theta)) \qquad U_i(\theta) = -\log(p(z_i|\theta))$$

If N is large, a single iteration of ULA may be expensive

- Remedy: use ideas from Stochastic Gradient Descent (SGD), i.e. do not use the full gradient but an unbiased random approximation

Langevin Dynamics – Bayesian inference

This gives rise to the Stochastic Gradient Langevin Synamis (SGLD) algorithm (Welling & Teh 2011)

$$\theta_{k+1} = \theta_k - \gamma \left(\nabla U_0(\theta_k) + \frac{N}{p} \sum_{i \in S_{k+1}} \nabla U_i(\theta_k) \right) + \sqrt{2\gamma} Z_{k+1}$$

Much smaller cost per iteration if p << N

- Again, simply a MAP **stochastic descent algorithm** with noise added at each iteration
- Theoretical convergence studied in Brosse et al. 2018

Langevin Dynamics – Bayesian inference

Assuming we have an estimate of the mode of the distribution, we can design a control variates version of SGLD

$$\theta_{k+1} = \theta_k - \gamma \left(\nabla U_0(\theta_k) - \nabla U_0(\theta^\star) + \frac{N}{p} \sum_{i \in S_{k+1}} \{\nabla U_i(\theta_k) - \nabla U_i(\theta^\star)\} \right) + \sqrt{2\gamma} Z_{k+1}$$

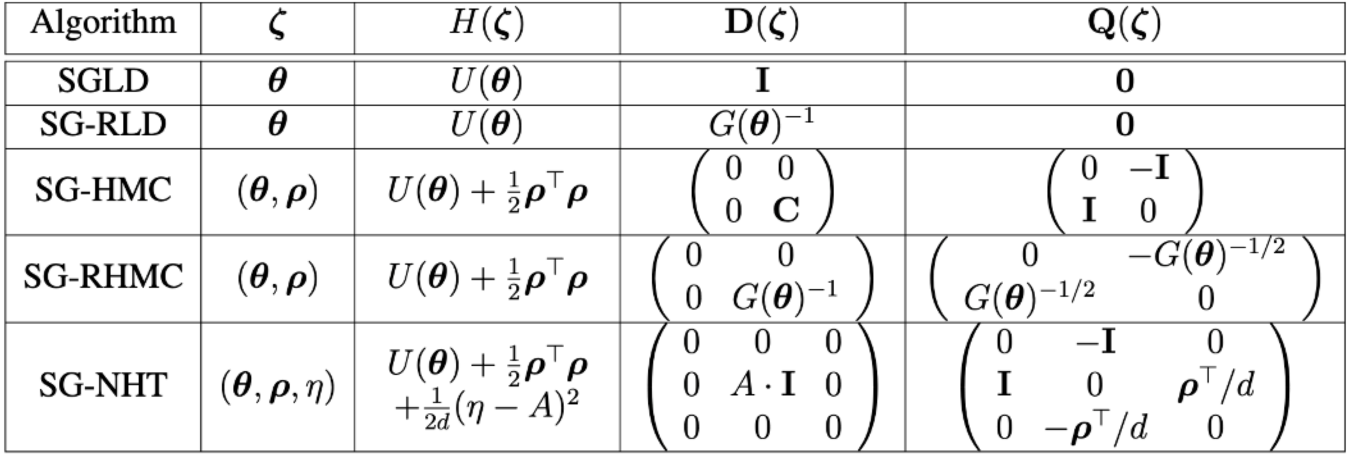
This is SGLD Control Variate (SGLDCV) or SGLD Fixed Point (SGLDFP), Dubey et al. 2016

- **Smaller burnin, faster convergence, but additional cost to get estimate of the mode**
  - In practice, very often a first SGD is launched
  - Theoretical convergence studied in Brosse et al. 2018
- Other ways to control variance via weighted sampling or stratified sampling (Nemeth & Fearnhead 2019)

Langevin Dynamics – More general framework

We can potentially add auxiliary variables to the SDE and end up with a general SDE (Nemeth & Fearnhead 2019)

$$\boldsymbol{\zeta}_{t+h} \approx \boldsymbol{\zeta}_t - \frac{h}{2}\left[(\mathbf{D}(\boldsymbol{\zeta}_t) + \mathbf{Q}(\boldsymbol{\zeta}_t))\nabla H(\boldsymbol{\zeta}_t) + \Gamma(\boldsymbol{\zeta}_t)\right] + \sqrt{h}\mathbf{Z}$$



Algorithm	$\boldsymbol{\zeta}$	$H(\boldsymbol{\zeta})$	$\mathbf{D}(\boldsymbol{\zeta})$	$\mathbf{Q}(\boldsymbol{\zeta})$
SGLD	$\boldsymbol{\theta}$	$U(\boldsymbol{\theta})$	$\mathbf{I}$	$\mathbf{0}$
SG-RLD	$\boldsymbol{\theta}$	$U(\boldsymbol{\theta})$	$G(\boldsymbol{\theta})^{-1}$	$\mathbf{0}$
SG-HMC	$(\boldsymbol{\theta}, \boldsymbol{\rho})$	$U(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{\rho}^\top\boldsymbol{\rho}$	$\begin{pmatrix} 0 & 0 \\ 0 & \mathbf{C} \end{pmatrix}$	$\begin{pmatrix} 0 & -\mathbf{I} \\ \mathbf{I} & 0 \end{pmatrix}$
SG-RHMC	$(\boldsymbol{\theta}, \boldsymbol{\rho})$	$U(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{\rho}^\top\boldsymbol{\rho}$	$\begin{pmatrix} 0 & 0 \\ 0 & G(\boldsymbol{\theta})^{-1} \end{pmatrix}$	$\begin{pmatrix} 0 & -G(\boldsymbol{\theta})^{-1/2} \\ G(\boldsymbol{\theta})^{-1/2} & 0 \end{pmatrix}$
SG-NHT	$(\boldsymbol{\theta}, \boldsymbol{\rho}, \eta)$	$U(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{\rho}^\top\boldsymbol{\rho}$ $+\frac{1}{2d}(\eta - A)^2$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & A\cdot\mathbf{I} & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & -\mathbf{I} & 0 \\ \mathbf{I} & 0 & \boldsymbol{\rho}^\top/d \\ 0 & -\boldsymbol{\rho}^\top/d & 0 \end{pmatrix}$

Riemannian SGLD (Fisher)

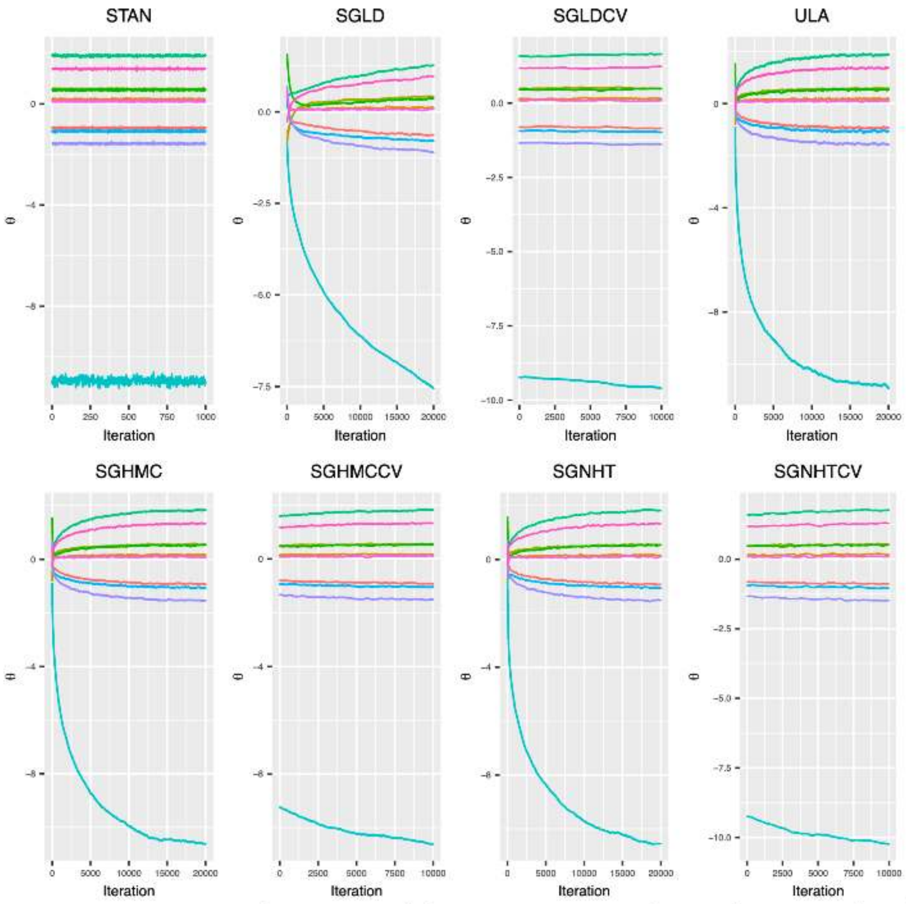
Hamiltonian MC

Riemannian Hamiltonian MC

Nose-Hoover thermostat

Nemeth & Fearnhead 2019

Langevin Dynamics – Example



Bayesian logistic regression in dimension 10

- ULA, SGHMC and SGHNT have similar convergence, but higher iteration cost than SGLD
- Control variate versions converge faster
  - But they need an initialization phase, whose cost is not accounted for here

Nemeth & Fearnhead 2019

Langevin Dynamics – Applications in ML/DL

ML examples

- Bayesian logistic regression d=123 (Welling & Teh 2011, …)
- Image denoising & deconvolution d=256*256 (Durmus et al. 2018)
- Regression d = 2 – 90 (Dubey et al. 2016)
- Matrix factorization d = 256 * 140 (Simsekli et al. 2016)
- …

DL examples

- Deep ensembles d= 100 – 600 (Lakshminarayanan et al. 2017)
- Weight uncertainty d up to 1200 (Li et al. 2016)
- ...

Langevin Dynamics – Potential applications in UQ ?

Sampling the full posterior for Gaussian Processes

- May be less expensive than MCMC and more scalable w.r.t. dimension
  - Quite easy to implement since we already have gradients of the log-likelihood in packages
- Easy to implement bound constraints or sparsity-inducing priors with MYULA
- Build upon previous SGD applied to GPs for large data sets (Filippone & Engler 2015, Yan et al. 2015)

Bayesian calibration of computer codes

- We very often use MCMC on a GP approximation of the computer code to sample from the calibration parameters
- But with just the GP derivatives (available in most packages), it is simple to use a LD algorithm to replace MCMC
- If the number of data to calibrate is large, can even think about using a SGD version

… ?

Langevin Dynamics – Software

R: sgmcmc (Baker et al. 2019)

- Based on TensorFlow
- SGLD, SGLDCV, SGHMC, SGHMCCV, SGNHT, SGNHTCV

C++/Armadillo: MCMClib (O’Hara, https://www.kthohr.com/mcmclib.html)

- (HMC), (RWMH), MALA

Python: Edward (Tran et al. 2016)

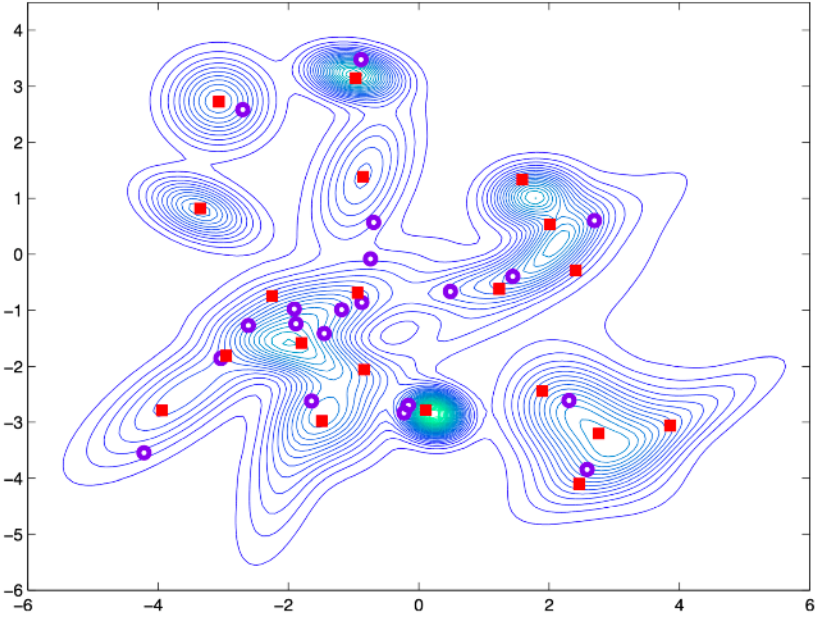
- (HMC), SGLD

2

(STEIN) KERNELS FOR SUBSAMPLING

Efficient subsampling

Goal: for a given probability distribution (or a large Monte-Carlo sample from it), find a small number of points which best represent it



Chen et al. 2012

Efficient subsampling

Goal: for a given probability distribution (or a large Monte-Carlo sample from it), find a small number of points which best represent it

Note: this is what we do with discrepancy for building space-filling designs (uniform distribution)

A lot of recent work dedicated to answer this question with kernels

- Ingredient 1: write the subsampling problem as an **optimization problem** = find the mixture of Diracs (i.e. the subsample) which has the smallest **distance** to the target probability distribution
- Ingredient 2: choose a distance to compare probability distributions = a **kernel-based distance**
- Ingredient 3: select an **optimization** algorithm
- Variety in ingredients 2 & 3 gave rise to several methodologies

Efficient subsampling – Ingredient 1 Kernel-based distance

Given two probability distributions and a RKHS with kernel k(.,.), we define the Maximum Mean Discrepancy (MMD)

$$\begin{aligned}\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) &= \left\|\int k(x,\cdot)d\mathbb{P}(x) - \int k(x,\cdot)d\mathbb{Q}(x)\right\|_{\mathcal{H}}^2 \\ &= \mathbb{E}_{X\sim\mathbb{P},X'\sim\mathbb{P}}\left[k(X,X')\right] + \mathbb{E}_{Y\sim\mathbb{Q},Y'\sim\mathbb{Q}}\left[k(Y,Y')\right] \\ &\quad -2\mathbb{E}_{X\sim\mathbb{P},Y\sim\mathbb{Q}}\left[k(X,Y)\right]\end{aligned}$$

So for subsampling the problem writes

$$\boxed{\underset{X_1,\ldots,X_n}{\mathrm{Argmin}}\ \mathrm{MMD}^2\left(\frac{1}{n}\sum_{i=1}^{n}\delta_{X_i},\mathbb{P}\right) = \underset{X_1,\ldots,X_n}{\mathrm{Argmax}}\ \sum_{i=1}^{n}\int k(X_i,u)d\mathbb{P}(u) - \frac{1}{2n}\sum_{i,j=1}^{n}k(X_i,X_j)}$$

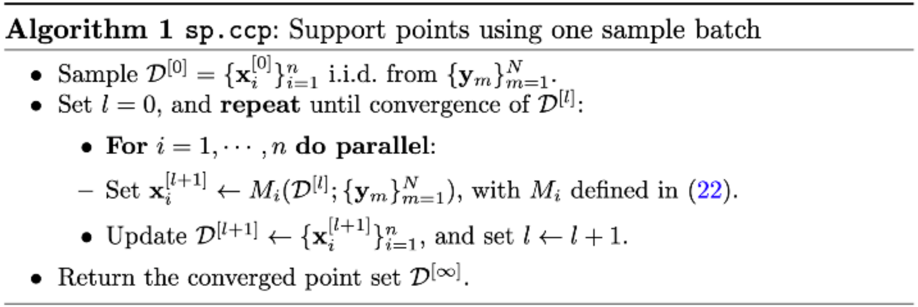
Efficient subsampling – Ingredient 2 Solving the optimization problem

Greedy approach: Kernel-herding (Chen et al. 2012)

$$\mathbf{x}_{T+1} = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}}\, \mathbb{E}_{\mathbf{x}'\sim p}[k(\mathbf{x},\mathbf{x}')] - \frac{1}{T+1}\sum_{t=1}^{T} k(\mathbf{x},\mathbf{x}_t)$$

- They use a universal kernel, e.g. Gaussian and Laplace

Convex-concave programming: Support points (Mak & Joseph 2018)



```
Algorithm 1 sp.ccp: Support points using one sample batch
  • Sample $\mathcal{D}^{[0]} = \{\mathbf{x}_i^{[0]}\}_{i=1}^n$ i.i.d. from $\{\mathbf{y}_m\}_{m=1}^N$.
  • Set $l = 0$, and repeat until convergence of $\mathcal{D}^{[l]}$:
      • For $i = 1, \cdots, n$ do parallel:
      – Set $\mathbf{x}_i^{[l+1]} \leftarrow M_i(\mathcal{D}^{[l]}; \{\mathbf{y}_m\}_{m=1}^N)$, with $M_i$ defined in (22).
      • Update $\mathcal{D}^{[l+1]} \leftarrow \{\mathbf{x}_i^{[l+1]}\}_{i=1}^n$, and set $l \leftarrow l + 1$.
  • Return the converged point set $\mathcal{D}^{[\infty]}$.
```

- They use a specific kernel (energy distance – distance correlation) and get a convex upper bound
- **Generalized in projected support points (Mak & Joseph 2017)**

Efficient subsampling – Example of usage

« Big-data reduction » (terminology from Mak & Joseph 2017)

- Target distribution is given as a large sample and we want to summarize it
- **Generate training/test samples**
- **Instead of propagating a full MCMC, use a smart subsample**

Efficient subsampling – Example of usage

« Big-data reduction » (terminology from Mak & Joseph 2017)

- Target distribution is given as a large sample and we want to summarize it
- **Generate training/test samples**
- **Instead of propagating a full MCMC, use a smart subsample**

But this implies we need to know the target distribution or have a sample

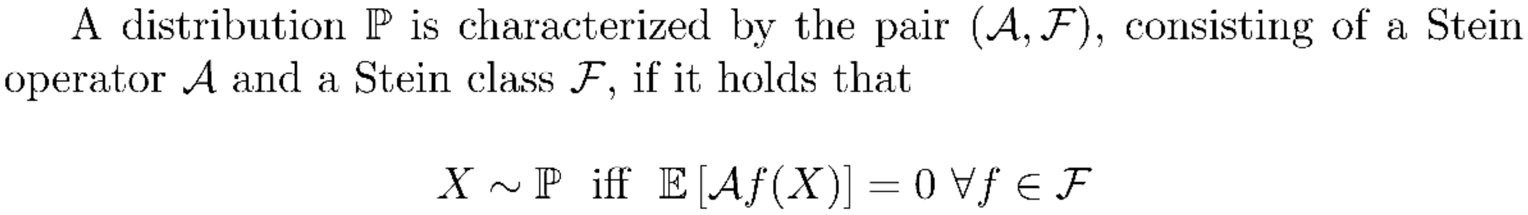
- Coming back to MCMC example
  - We could use a generated sample (but this is expensive …)
  - … but we almost know the target distribution (up to a constant)

$$\underset{X_1,\dots,X_n}{\text{Argmin}}\ \text{MMD}^2\left(\frac{1}{n}\sum_{i=1}^{n}\delta_{X_i},\mathbb{P}\right)=\underset{X_1,\dots,X_n}{\text{Argmax}}\ \sum_{i=1}^{n}\boxed{\int k(X_i,u)d\mathbb{P}(u)}-\frac{1}{2n}\sum_{i,j=1}^{n}k(X_i,X_j)$$

- **Hint: what if we can come up with a zero-mean kernel, which is defined via the unnormalized target distribution ?**
- **Answer: Stein kernels**

Efficient subsampling – Stein kernels

Definition (Stein Characterization)

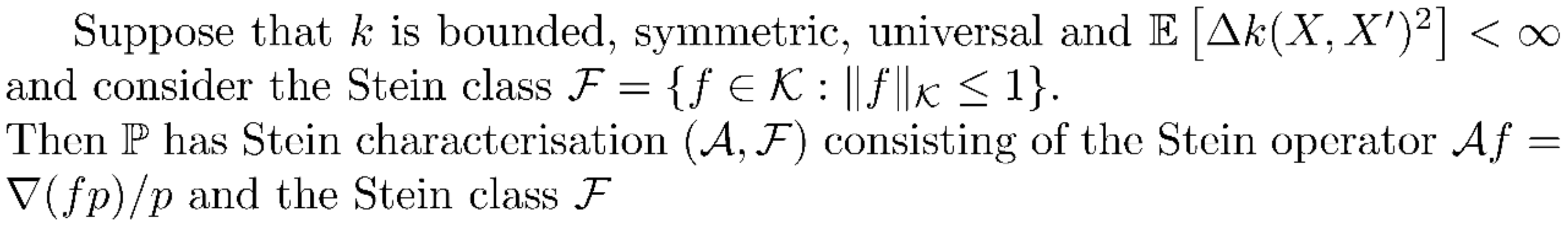


A distribution $\mathbb{P}$ is characterized by the pair $(\mathcal{A}, \mathcal{F})$, consisting of a Stein operator $\mathcal{A}$ and a Stein class $\mathcal{F}$, if it holds that

$$X \sim \mathbb{P} \ \text{ iff } \ \mathbb{E}\left[\mathcal{A}f(X)\right] = 0 \ \forall f \in \mathcal{F}$$

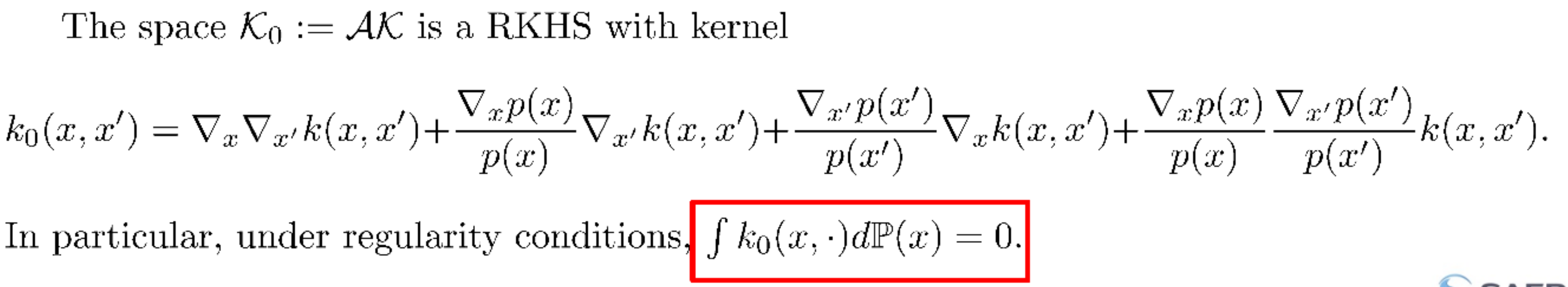
Efficient subsampling – Stein kernels

In particular for RKHS, we have the following theorem (Chwialkowski et al. 2016)



Suppose that k is bounded, symmetric, universal and $\mathbb{E}\left[\Delta k(X,X')^2\right] < \infty$ and consider the Stein class $\mathcal{F} = \{f \in \mathcal{K} : \|f\|_{\mathcal{K}} \leq 1\}$.
Then $\mathbb{P}$ has Stein characterisation $(\mathcal{A},\mathcal{F})$ consisting of the Stein operator $\mathcal{A}f = \nabla(fp)/p$ and the Stein class $\mathcal{F}$

Which can be used in practive via the theorem (Oates, Girolami, Chopin 2017)



The space $\mathcal{K}_0 := \mathcal{A}\mathcal{K}$ is a RKHS with kernel

$$k_0(x,x') = \nabla_x\nabla_{x'}k(x,x') + \frac{\nabla_x p(x)}{p(x)}\nabla_{x'}k(x,x') + \frac{\nabla_{x'}p(x')}{p(x')}\nabla_x k(x,x') + \frac{\nabla_x p(x)}{p(x)}\frac{\nabla_{x'}p(x')}{p(x')}k(x,x').$$

In particular, under regularity conditions, $\boxed{\int k_0(x,\cdot)d\mathbb{P}(x) = 0.}$

Efficient subsampling – Stein kernels

This leads to replacing the MMD by the kernel Stein discrepancy (KSD)

$$\underset{X_1,\ldots,X_n}{\mathrm{Argmin}}\ D^2_{K_0,\mathbb{P}}\left(X_1,\ldots,X_n\right) = \frac{1}{n^2}\sum_{i,j=1}^{n} k_0(X_i, X_j)$$

Greedy algorithm proposed in Chen et al. 2018 under the name « Stein points »

MCMC version in Chen et al. 2019: « Stein point MCMC »

Efficient subsampling – Stein kernels potential in UQ

« Big-data reduction » (terminology from Mak & Joseph 2017)

- Target distribution is given as a large sample and we want to summarize it
- Generate training/test samples
- Instead of propagating a full MCMC, use a smart subsample

- **Replace the MCMC sampling + subsampling by directly using a Stein kernel inside the sampler**
  - Stein Variational Gradient Descent (Liu & Wang 2016, Liu 2017) + 2nd order (Detommaso et al. 2018) + high-dimension (Chen et al. 2019)
  - Measuring sample quality (Gorham et al. 2019)
- Importance Sampling for black-box (Liu & Li 2017)
- Post-hoc correction of MCMC sample (Hodgkinson et al. 2020)
- Random Feature Stein Discrepancies (Huggins & Mackey 2018)

Thank you for your attention

References (1/2)

Baker, J., Fearnhead, P., Fox, E. B., & Nemeth, C. (2019). sgmcmc: An R Package for Stochastic Gradient Markov Chain Monte Carlo. *Journal of Statistical Software*, *91*(1), 1-27.

Brosse, N., Durmus, A., & Moulines, E. (2018). The promises and pitfalls of stochastic gradient Langevin dynamics. In *Advances in Neural Information Processing Systems* (pp. 8268-8278).

Chen, P., Wu, K., Chen, J., O'Leary-Roseberry, T., & Ghattas, O. (2019). Projected Stein variational Newton: A fast and scalable Bayesian inference method in high dimensions. In Advances in Neural Infor

Chen, W. Y., Barp, A., Briol, F. X., Gorham, J., Girolami, M., Mackey, L., & Oates, C. (2019). Stein point markov chain monte carlo. *arXiv preprint arXiv:1905.03673*.

Chen, W. Y., Mackey, L., Gorham, J., Briol, F. X., & Oates, C. J. (2018). Stein points. *arXiv preprint arXiv:1803.10161*.

Chen, Y., Welling, M., & Smola, A. (2012). Super-samples from kernel herding. *arXiv preprint arXiv:1203.3472*.

Chwialkowski, K., Strathmann, H., & Gretton, A. (2016, June). A kernel test of goodness of fit. JMLR: Workshop and Conference Proceedings.

Detommaso, G., Cui, T., Marzouk, Y., Spantini, A., & Scheichl, R. (2018). A Stein variational Newton method. In Advances in Neural Information Processing Systems (pp. 9169-9179).

Dubey, K. A., Reddi, S. J., Williamson, S. A., Poczos, B., Smola, A. J., & Xing, E. P. (2016). Variance reduction in stochastic gradient Langevin dynamics. In *Advances in neural information processing systems* (pp. 1154-1162).

Durmus, A., Moulines, E., & Pereyra, M. (2018). Efficient bayesian computation by proximal markov chain monte carlo: when langevin meets moreau. *SIAM Journal on Imaging Sciences*, *11*(1), 473-506.

Durmus, A., & Moulines, E. (2017). Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *The Annals of Applied Probability*, *27*(3), 1551-1587.

Filippone, M., & Engler, R. (2015). Enabling scalable stochastic gradient-based inference for Gaussian processes by employing the Unbiased LInear System SolvEr (ULISSE). *arXiv preprint arXiv:1501.05427*.

Gorham, J., Duncan, A. B., Vollmer, S. J., & Mackey, L. (2019). Measuring sample quality with diffusions. The Annals of Applied Probability, 29(5), 2884-2928.

Hodgkinson, L., Salomone, R., & Roosta, F. (2020). The reproducing Stein kernel approach for post-hoc corrected sampling. arXiv preprint arXiv:2001.09266.

References (2/2)

Huggins, J., & Mackey, L. (2018). Random feature stein discrepancies. In Advances in Neural Information Processing Systems (pp. 1899-1909).

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In Advances in neural information processing systems (pp. 6402-6413).

Li, C., Stevens, A., Chen, C., Pu, Y., Gan, Z., & Carin, L. (2016). Learning weight uncertainty with stochastic gradient mcmc for shape classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5666-5675).

Liu, Q. (2017). Stein variational gradient descent as gradient flow. In Advances in neural information processing systems (pp. 3115-3123).

Liu, Q., & Lee, J. (2017, April). Black-box Importance Sampling. In Artificial Intelligence and Statistics (pp. 952-961).

Liu, Q., & Wang, D. (2016). Stein variational gradient descent: A general purpose bayesian inference algorithm. In Advances in neural information processing systems (pp. 2378-2386).

Mak, S., & Joseph, V. R. (2018). Support points. *The Annals of Statistics*, *46*(6A), 2562-2592.

Mak, S., & Joseph, V. R. (2017). Projected support points: a new method for high-dimensional data reduction. *arXiv preprint arXiv:1708.06897*.

Nemeth, C., & Fearnhead, P. (2019). Stochastic gradient Markov chain Monte Carlo. *arXiv preprint arXiv:1907.06986*.

Oates, C. J., Girolami, M., & Chopin, N. (2017). Control functionals for Monte Carlo integration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *79*(3), 695-718.

Simsekli, U., Badeau, R., Cemgil, T., & Richard, G. (2016, June). Stochastic quasi-newton langevin monte carlo.

Tran, D., Kucukelbir, A., Dieng, A. B., Rudolph, M., Liang, D., & Blei, D. M. (2016). Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*.

Yan, X., Xie, B., Song, L., Boots, B., & EDU, G. (2015). Large-scale Gaussian process regression via doubly stochastic gradient descent. In *The ICML Workshop on Large-Scale Kernel Learning*.

Welling, M., & Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 681-688).