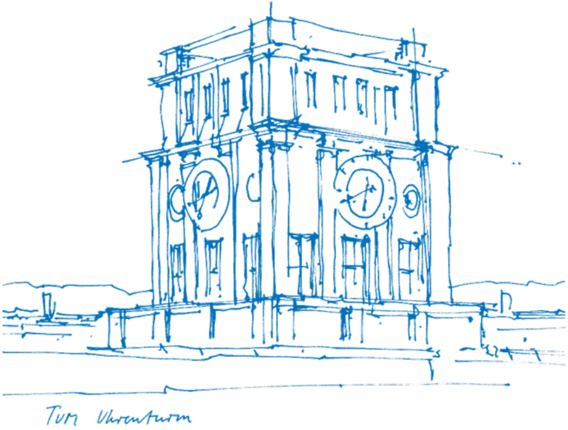
Chair of Scientific Computing
TUM School of Computation, Information and Technology
Technical University of Munich

Multi-fidelity No-U-Turn Sampling

SIAM CSE 2023



Kislaya Ravi

Chair of Scientific Computing
Technical University of Munich

February 28th, 2023

Outline

Bayesian Inverse

- Let $\Theta \in \mathbb{R}^d$ and $\mathbf{Y} \in \mathbb{R}^m$ represent the parameter and the data space respectively.

Bayesian Inverse

- Let $\Theta \in \mathbb{R}^d$ and $\mathbf{Y} \in \mathbb{R}^m$ represent the parameter and the data space respectively.
- For $\theta \in \Theta$, the noisy observations $y \in \mathbf{Y}$ can be expressed as:

$$y = f(\theta) + \eta$$

  where η is the noise term.

Bayesian Inverse

- Let $\Theta \in \mathbb{R}^d$ and $\mathbf{Y} \in \mathbb{R}^m$ represent the parameter and the data space respectively.
- For $\theta \in \Theta$, the noisy observations $y \in \mathbf{Y}$ can be expressed as:

$$y = f(\theta) + \eta$$

  where η is the noise term.
- Based on noise term, calculate the likelihood ($p(y|\theta)$). For $\eta \sim \mathcal{N}(0, \Gamma)$

$$p(y|\theta) = \exp\left(-\frac{1}{2}||\Gamma^{-1/2}(y - f(\theta))||^2\right)$$

- Using Bayes theorem, evaluate the posterior ($p(\theta|y)$)

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

Bayesian Inverse

- Let $\Theta \in \mathbb{R}^d$ and $\mathbf{Y} \in \mathbb{R}^m$ represent the parameter and the data space respectively.
- For $\theta \in \Theta$, the noisy observations $y \in \mathbf{Y}$ can be expressed as:

$$y = f(\theta) + \eta$$

  where η is the noise term.
- Based on noise term, calculate the likelihood ($p(y|\theta)$). For $\eta \sim \mathcal{N}(0, \Gamma)$

$$p(y|\theta) = \exp\left(-\frac{1}{2}||\Gamma^{-1/2}(y - f(\theta))||^2\right)$$

- Using Bayes theorem, evaluate the posterior ($p(\theta|y)$)

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

- **TODO:** Sample from $p(\theta|y)$

MCMC computationaly expensive model

- Sample from a density function which is computationally expensive.
- Becomes challenging for complicated domain/ high-dimensional problems

MCMC computationaly expensive model

- Sample from a density function which is computationally expensive.
- Becomes challenging for complicated domain/ high-dimensional problems
- Gradient based methods (HMC, NUTS etc.) can help
  - Need Gradient
  - Gradient evaluation is needed at multiple points $\Longrightarrow$ Infeasible for computationally expensive models

MCMC computationaly expensive model

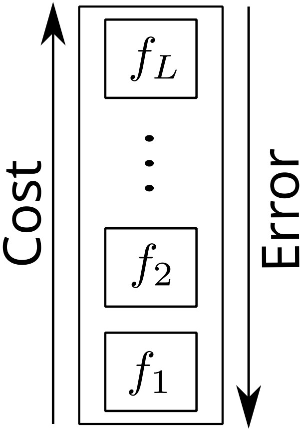
- Sample from a density function which is computationally expensive.
- Becomes challenging for complicated domain/ high-dimensional problems
- Gradient based methods (HMC, NUTS etc.) can help
  - Need Gradient
  - Gradient evaluation is needed at multiple points $\Longrightarrow$ Infeasible for computationally expensive models
- **Task:** Alleviate this issue using *Multi-fidelity*.

Multi-fidelity

- Supoose we are given ordered set of models as:

$$F = \{f_1, f_2, \cdots, f_L\}$$

where, $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ is the i^{th} model



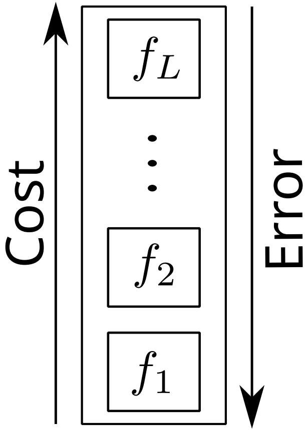
Multi-fidelity

- Supoose we are given ordered set of models as:

$$F = \{f_1, f_2, \cdots, f_L\}$$

where, $f_i : \mathbb{R}^d \to \mathbb{R}$ is the i^{th} model

- The models are ordered in:
  - Ascending order of computational intensity or cost of getting results or
  - Decreasing error

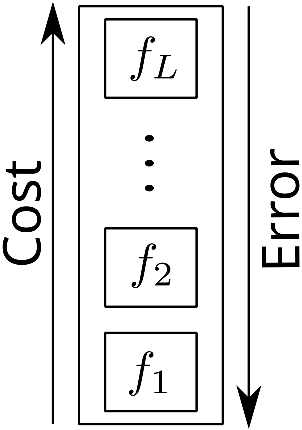


Multi-fidelity

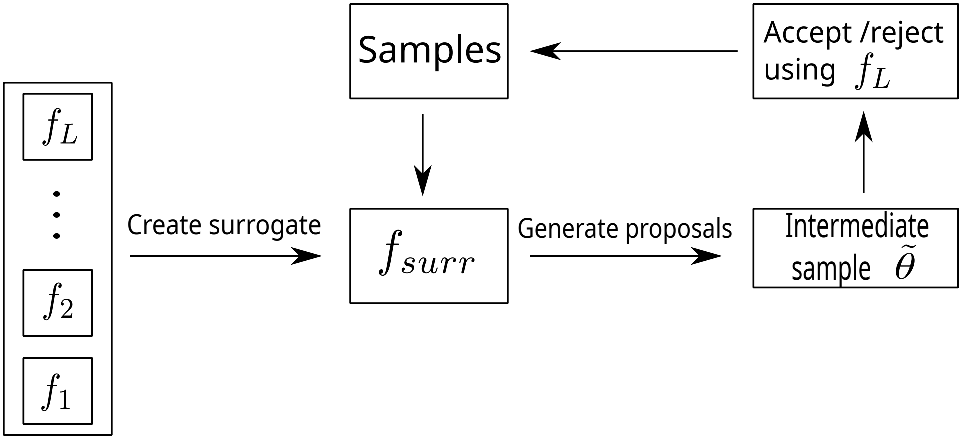
- Supoose we are given ordered set of models as:

$$F = \{f_1, f_2, \cdots, f_L\}$$

  where, $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ is the i^{th} model
- The models are ordered in:
  - Ascending order of computational intensity or cost of getting results or
  - Decreasing error
- In multi-fidelity methods, we try to solve given problem in hand by transferring maximum workload to lower fidelity models



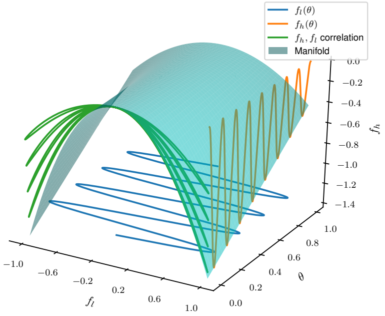
Flowchart


Samples
Accept /reject
using f_L
f_L
f_2
f_1
Create surrogate
f_{surr}
Generate proposals
Intermediate
sample $\tilde{\theta}$

Outline

Multi-fidelity implementation[1]

- High fidelity function contains features from the low-fidelity function and some additional new features.

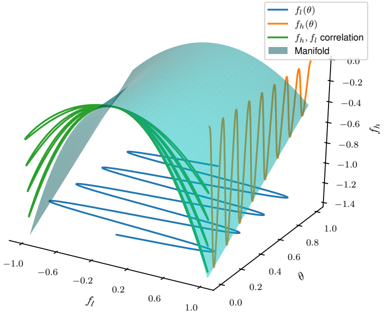


[1] Perdikaris, Paris, et al. "Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473.2198 (2017): 20160751.

Multi-fidelity implementation[1]

- High fidelity function contains features from the low-fidelity function and some additional new features.
- Write high-fidelity function as composite function

$$f_h(\theta) = g(f_l(\theta), \theta)$$



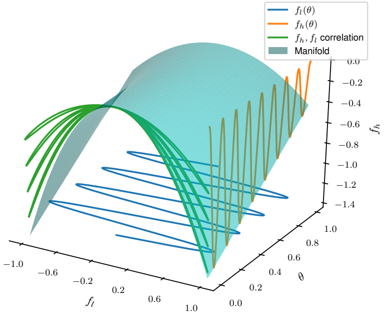
[1] Perdikaris, Paris, et al. "Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473.2198 (2017): 20160751.

Multi-fidelity implementation[1]

- High fidelity function contains features from the low-fidelity function and some additional new features.
- Write high-fidelity function as composite function

$$f_h(\theta) = g(f_l(\theta), \theta)$$

- Some information is carried over from the low-fidelity function



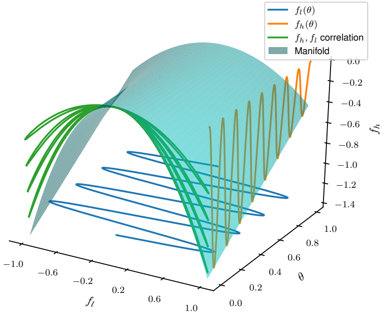
[1] Perdikaris, Paris, et al. "Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473.2198 (2017): 20160751.

Multi-fidelity implementation[1]

- High fidelity function contains features from the low-fidelity function and some additional new features.
- Write high-fidelity function as composite function

$$f_h(\theta) = g(f_l(\theta), \theta)$$

- Some information is carried over from the low-fidelity function
- In this work, we use Gaussian Process for g

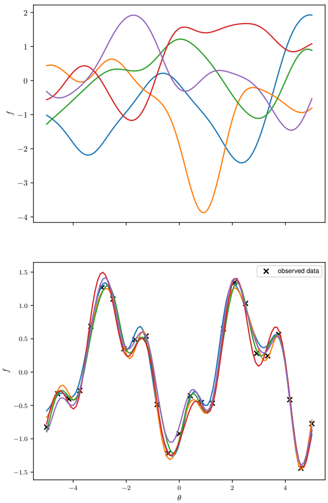


[1] Perdikaris, Paris, et al. "Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473.2198 (2017): 20160751.

Gaussian Process [2]

- Gaussian Process is a bayesian model
- Assume prior

$$f \sim \mathcal{N}(0, K)$$



[2] Rasmussen, Carl Edward. "Gaussian processes in machine learning."

Gaussian Process [2]

- Gaussian Process is a bayesian model
- Assume prior

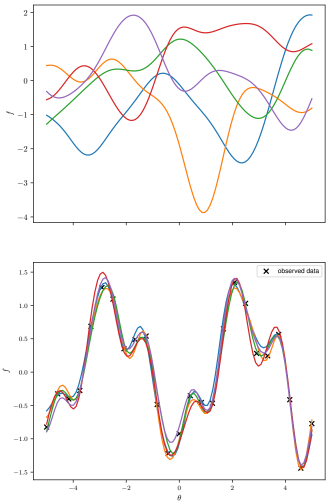
$$f \sim \mathcal{N}(0, K)$$

- Prediction at X_* after observing data (X, y) with noise σ^2

$$p(f_* | y, \theta, \theta_*) \sim \mathcal{N}(\hat{\mu}, \hat{\Sigma})$$

$$\hat{\mu} = K(\theta_*, \theta)[K(\theta, \theta) + \sigma^2 I_N]^{-1} y$$

$$\hat{\Sigma} = K(\theta_*, \theta_*) - K(\theta_*, \theta)[K(\theta, \theta) + \sigma^2 I_N]^{-1} K(\theta, \theta_*)$$



[2] Rasmussen, Carl Edward. "Gaussian processes in machine learning."

Gaussian Process [2]

- Gaussian Process is a bayesian model
- Assume prior

$$f \sim \mathcal{N}(0, K)$$

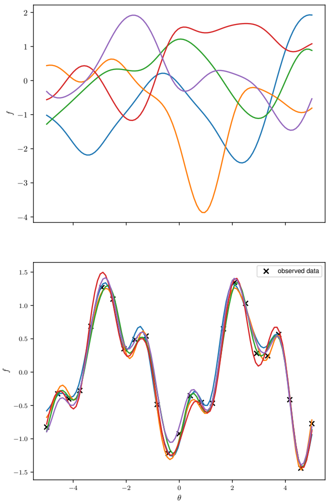
- Prediction at X_* after observing data (X, y) with noise σ^2

$$p(f_*|y, \theta, \theta_*) \sim \mathcal{N}(\hat{\mu}, \hat{\Sigma})$$

$$\hat{\mu} = K(\theta_*, \theta)[K(\theta, \theta) + \sigma^2 I_N]^{-1} y$$

$$\hat{\Sigma} = K(\theta_*, \theta_*) - K(\theta_*, \theta)[K(\theta, \theta) + \sigma^2 I_N]^{-1} K(\theta, \theta_*)$$

- Kernel hyperparameters (λ) can be trained by maximizing likelihood



[2] Rasmussen, Carl Edward. ”Gaussian processes in machine learning.”

Multi-fidelity in GP implementation

- Expand the kernel [1]:

$$K(\theta, \theta') = K_\delta(\theta, \theta'; \lambda_1) + K_\rho(\theta, \theta'; \lambda_2) K_f(f_l(\theta), f_l(\theta'); \lambda_3)$$

[1] Perdikaris, Paris, et al. "Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473.2198 (2017): 20160751.

[3] Lee, Seungjoon, et al. "Linking Gaussian process regression with data-driven manifold embeddings for nonlinear data fusion." Interface focus 9.3 (2019): 20180083.

Multi-fidelity in GP implementation

- Expand the kernel [1]:

$$K(\theta, \theta') = K_\delta(\theta, \theta'; \lambda_1) + K_\rho(\theta, \theta'; \lambda_2) K_f(f_l(\theta), f_l(\theta'); \lambda_3)$$

- Variation to include derivative term by using lag term to mimic derivative [3]:

$$f_h(\theta) = g(f_l(\theta), f_l(\theta - \tau), f_l(\theta + \tau), \theta)$$

[1] Perdikaris, Paris, et al. "Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473.2198 (2017): 20160751.

[3] Lee, Seungjoon, et al. "Linking Gaussian process regression with data-driven manifold embeddings for nonlinear data fusion." Interface focus 9.3 (2019): 20180083.

Multi-fidelity in GP implementation

- Expand the kernel [1]:

$$K(\theta, \theta') = K_\delta(\theta, \theta'; \lambda_1) + K_\rho(\theta, \theta'; \lambda_2) K_f(f_l(\theta), f_l(\theta'); \lambda_3)$$

- Variation to include derivative term by using lag term to mimic derivative [3]:

$$f_h(\theta) = g(f_l(\theta), f_l(\theta - \tau), f_l(\theta + \tau), \theta)$$

- Adaptively add points where gain of information ($\mathcal{I}$) is maximized, which corresponds to finding maximum posterior variance:

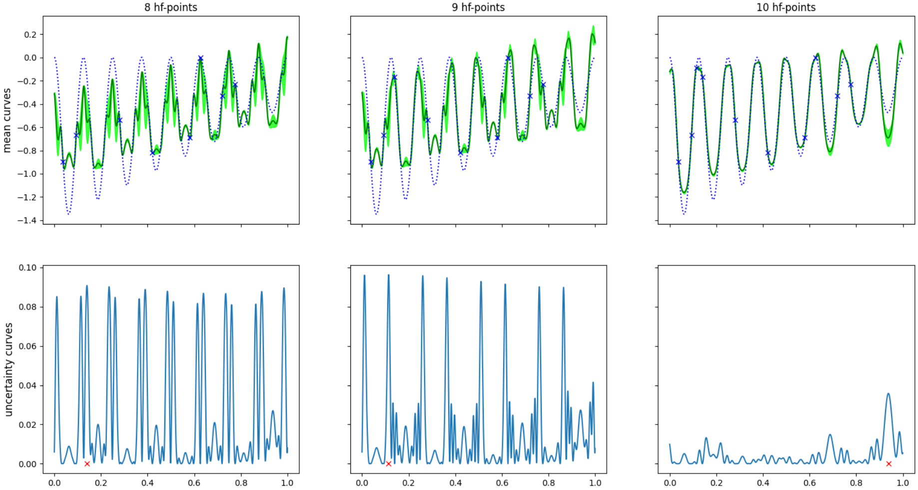
$$X_{new} = \arg\max_{\theta \in \Omega} \mathcal{I} = \arg\max_{\theta \in \Omega} \hat{\Sigma}$$

---

[1] Perdikaris, Paris, et al. ”Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling.” Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473.2198 (2017): 20160751.

[3] Lee, Seungjoon, et al. ”Linking Gaussian process regression with data-driven manifold embeddings for nonlinear data fusion.” Interface focus 9.3 (2019): 20180083.

Example: Adaptivity



Outline

Hamilton Monte Carlo [4]

- Gradient based method to incorporate some geometrical information.

[4] R. Neal. ”Handbook of Markov Chain Monte Carlo”, chapter 5: MCMC Using Hamiltonian Dynamics. CRC Press, 2011.

Hamilton Monte Carlo [4]

- Gradient based method to incorporate some geometrical information.
- Introduce a momentum term r representing kinetic energy $(K(r) = \frac{1}{2} r \cdot r)$ and represent log of the target density $(\mathcal{L}(\theta) = \log p(\theta))$. The Hamiltonian can be defined as $H(\theta, r) = K(r) - \mathcal{L}(\theta)$

[4] R. Neal. "Handbook of Markov Chain Monte Carlo", chapter 5: MCMC Using Hamiltonian Dynamics. CRC Press, 2011.

Hamilton Monte Carlo [4]

- Gradient based method to incorporate some geometrical information.
- Introduce a momentum term r representing kinetic energy $(K(r) = \frac{1}{2} r \cdot r)$ and represent log of the target density $(\mathcal{L}(\theta) = \log p(\theta))$. The Hamiltonian can be defined as $H(\theta, r) = K(r) - \mathcal{L}(\theta)$
- Sample from the joint canonical distribution $p(\theta, r) \propto \exp(-H(\theta, r))$

[4] R. Neal. "Handbook of Markov Chain Monte Carlo", chapter 5: MCMC Using Hamiltonian Dynamics. CRC Press, 2011.

Hamilton Monte Carlo [4]

- Gradient based method to incorporate some geometrical information.
- Introduce a momentum term r representing kinetic energy $(K(r) = \frac{1}{2}r \cdot r)$ and represent log of the target density $(\mathcal{L}(\theta) = \log p(\theta))$. The Hamiltonian can be defined as $H(\theta, r) = K(r) - \mathcal{L}(\theta)$
- Sample from the joint canonical distribution $p(\theta, r) \propto \exp(-H(\theta, r))$
- For the $(i+1)^{th}$ sample:
  - Randomly sample $r \sim \mathcal{N}(0, \mathbb{I}_d)$
  - Solve the Hamiltonian system for some time steps to propose a new point $(\tilde{\theta}, \tilde{r})$
  - Accept/Reject based on Metropolis-Hasting criterion $\alpha = \min\left[0, H(\theta, r) - H(\tilde{\theta}, \tilde{r})\right]$

[4] R. Neal. "Handbook of Markov Chain Monte Carlo", chapter 5: MCMC Using Hamiltonian Dynamics. CRC Press, 2011.

Hamilton Monte Carlo [4]

- Gradient based method to incorporate some geometrical information.
- Introduce a momentum term r representing kinetic energy $(K(r) = \frac{1}{2}r \cdot r)$ and represent log of the target density $(\mathcal{L}(\theta) = \log p(\theta))$. The Hamiltonian can be defined as $H(\theta, r) = K(r) - \mathcal{L}(\theta)$
- Sample from the joint canonical distribution $p(\theta, r) \propto \exp(-H(\theta, r))$
- For the $(i+1)^{th}$ sample:
  - Randomly sample $r \sim \mathcal{N}(0, \mathbb{I}_d)$
  - Solve the Hamiltonian system for some time steps to propose a new point $(\tilde{\theta}, \tilde{r})$
  - Accept/Reject based on Metropolis-Hasting criterion $\alpha = \min\left[0, H(\theta, r) - H(\tilde{\theta}, \tilde{r})\right]$
- Issues:
  - What is the time integration technique ? $\rightarrow$ Leap-frog method

[4] R. Neal. "Handbook of Markov Chain Monte Carlo", chapter 5: MCMC Using Hamiltonian Dynamics. CRC Press, 2011.

Hamilton Monte Carlo [4]

- Gradient based method to incorporate some geometrical information.
- Introduce a momentum term r representing kinetic energy $(K(r) = \frac{1}{2} r \cdot r)$ and represent log of the target density $(\mathcal{L}(\theta) = \log p(\theta))$. The Hamiltonian can be defined as $H(\theta, r) = K(r) - \mathcal{L}(\theta)$
- Sample from the joint canonical distribution $p(\theta, r) \propto \exp(-H(\theta, r))$
- For the $(i+1)^{th}$ sample:
  - Randomly sample $r \sim \mathcal{N}(0, \mathbb{I}_d)$
  - Solve the Hamiltonian system for some time steps to propose a new point $(\tilde{\theta}, \tilde{r})$
  - Accept/Reject based on Metropolis-Hasting criterion $\alpha = \min\left[0, H(\theta, r) - H(\tilde{\theta}, \tilde{r})\right]$
- Issues:
  - What is the time integration technique ? $\rightarrow$ Leap-frog method
  - What should be the step size? $\rightarrow$ Dual Averaging

[4] R. Neal. "Handbook of Markov Chain Monte Carlo", chapter 5: MCMC Using Hamiltonian Dynamics. CRC Press, 2011.

Hamilton Monte Carlo [4]

- Gradient based method to incorporate some geometrical information.
- Introduce a momentum term r representing kinetic energy $(K(r) = \frac{1}{2}r \cdot r)$ and represent log of the target density $(\mathcal{L}(\theta) = \log p(\theta))$. The Hamiltonian can be defined as $H(\theta, r) = K(r) - \mathcal{L}(\theta)$
- Sample from the joint canonical distribution $p(\theta, r) \propto \exp(-H(\theta, r))$
- For the $(i+1)^{th}$ sample:
  - Randomly sample $r \sim \mathcal{N}(0, \mathbb{I}_d)$
  - Solve the Hamiltonian system for some time steps to propose a new point $(\tilde{\theta}, \tilde{r})$
  - Accept/Reject based on Metropolis-Hasting criterion $\alpha = \min\left[0, H(\theta, r) - H(\tilde{\theta}, \tilde{r})\right]$
- Issues:
  - What is the time integration technique ? $\rightarrow$ Leap-frog method
  - What should be the step size? $\rightarrow$ Dual Averaging
  - How long should we perform the fictious time integration?

[4] R. Neal. "Handbook of Markov Chain Monte Carlo", chapter 5: MCMC Using Hamiltonian Dynamics. CRC Press, 2011.

No-U-Turn Sampling [5]

- Stopping criterion : Stop fictious time stepping when *U-turn* is observed:

$$(\theta - \tilde{\theta}) \cdot \tilde{r} < 0$$

[5] Hoffman, Matthew D., and Andrew Gelman. "The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." J. Mach. Learn. Res. 15.1 (2014): 1593-1623.

No-U-Turn Sampling [5]

- Stopping criterion : Stop fictious time stepping when *U-turn* is observed:

$$(\theta - \tilde{\theta}) \cdot \tilde{r} < 0$$

- Sample in both directions of the momentum (r and $-r$) by building a balanced tree and avoiding repetitive calculations.

[5] Hoffman, Matthew D., and Andrew Gelman. ”The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo.” J. Mach. Learn. Res. 15.1 (2014): 1593-1623.

No-U-Turn Sampling [5]

- Stopping criterion : Stop fictious time stepping when *U-turn* is observed:

$$(\theta - \tilde{\theta}) \cdot \tilde{r} < 0$$

- Sample in both directions of the momentum (r and $-r$) by building a balanced tree and avoiding repetitive calculations.
- Select the next point using slice sampling.

[5] Hoffman, Matthew D., and Andrew Gelman. "The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." J. Mach. Learn. Res. 15.1 (2014): 1593-1623.

No-U-Turn Sampling [5]

- Stopping criterion : Stop fictious time stepping when *U-turn* is observed:

$$(\theta - \tilde{\theta}) \cdot \tilde{r} < 0$$

- Sample in both directions of the momentum (r and $-r$) by building a balanced tree and avoiding repetitive calculations.
- Select the next point using slice sampling.
- For the $(i+1)^{th}$ sample :
  - Randomly sample $r \sim \mathcal{N}(0, \mathbb{I}_d)$
  - Draw a number from uniform distribution $\Delta \sim \mathcal{U}[0, p(\theta_i, p)]$
  - Solve the Hamiltonian system until U-turn and create a set of explored states.
  - Select the states that satisfy the criterion $\exp(H(\theta', r')) < \Delta$
  - Select one of the states from the above based on uniform distribution which become next sample.

[5] Hoffman, Matthew D., and Andrew Gelman. "The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." J. Mach. Learn. Res. 15.1 (2014): 1593-1623.

Multi-fidelity No-U-Turn Sampling

- We can directly sample from the multi-fidelity surrogate
  - Surrogate is cheap to evaluate
  - Gradient is available

[6] Christen, J. Andrés, and Colin Fox. "Markov chain Monte Carlo using an approximation." Journal of Computational and Graphical statistics 14.4 (2005): 795-810.

Multi-fidelity No-U-Turn Sampling

- We can directly sample from the multi-fidelity surrogate
  - Surrogate is cheap to evaluate
  - Gradient is available
- But, the samples obtained the not invariant for the highest fidelity models.

[6] Christen, J. Andrés, and Colin Fox. "Markov chain Monte Carlo using an approximation." Journal of Computational and Graphical statistics 14.4 (2005): 795-810.

Multi-fidelity No-U-Turn Sampling

- We can directly sample from the multi-fidelity surrogate
  - Surrogate is cheap to evaluate
  - Gradient is available
- But, the samples obtained the not invariant for the highest fidelity models.
- We follow the approach of Delayed acceptance [6]

[6] Christen, J. Andrés, and Colin Fox. "Markov chain Monte Carlo using an approximation." Journal of Computational and Graphical statistics 14.4 (2005): 795-810.

Multi-fidelity No-U-Turn Sampling

- Notation:
  - $p_s(\theta, r)$: Canonical density generated using the surrogate.
  - $\pi_L(\theta)$: Target density function generated using highest fidelity model.

Multi-fidelity No-U-Turn Sampling

- Notation:
  - $p_s(\theta, r)$: Canonical density generated using the surrogate.
  - $\pi_L(\theta)$: Target density function generated using highest fidelity model.
- For the $(i+1)^{th}$ sample:
  - Randomly sample $r \sim \mathcal{N}(0, \mathbb{I}_d)$

Multi-fidelity No-U-Turn Sampling

- Notation:
  - $p_s(\theta, r)$: Canonical density generated using the surrogate.
  - $\pi_L(\theta)$: Target density function generated using highest fidelity model.
- For the $(i+1)^{th}$ sample:
  - Randomly sample $r \sim \mathcal{N}(0, \mathbb{I}_d)$
  - Generate a proposal using NUTS $(\tilde{\theta}, \tilde{r})$

Multi-fidelity No-U-Turn Sampling

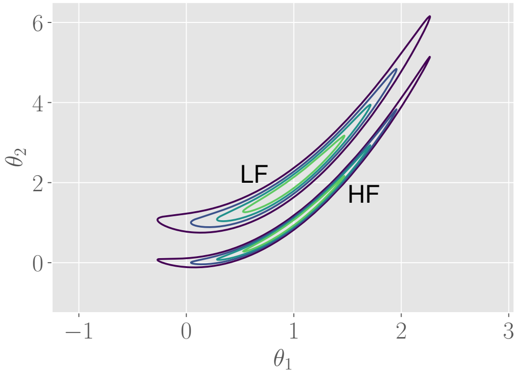
- Notation:
  - $p_s(\theta, r)$: Canonical density generated using the surrogate.
  - $\pi_L(\theta)$: Target density function generated using highest fidelity model.
- For the $(i+1)^{th}$ sample:
  - Randomly sample $r \sim \mathcal{N}(0, \mathbb{I}_d)$
  - Generate a proposal using NUTS $(\tilde{\theta}, \tilde{r})$
  - Accept/Reject based using delayed rejection

$$\alpha_{\text{MFNUTS}}(\tilde{\theta}|\theta) = \min\left\{1, \frac{\min\left\{1, \frac{p_s(\theta,r)}{p_s(\tilde{\theta},\tilde{r})}\right\} \pi_L(\tilde{\theta})}{\min\left\{1, \frac{p_s(\tilde{\theta},\tilde{r})}{p_s(\theta,r)}\right\} \pi_L(\theta)}\right\}$$

Outline

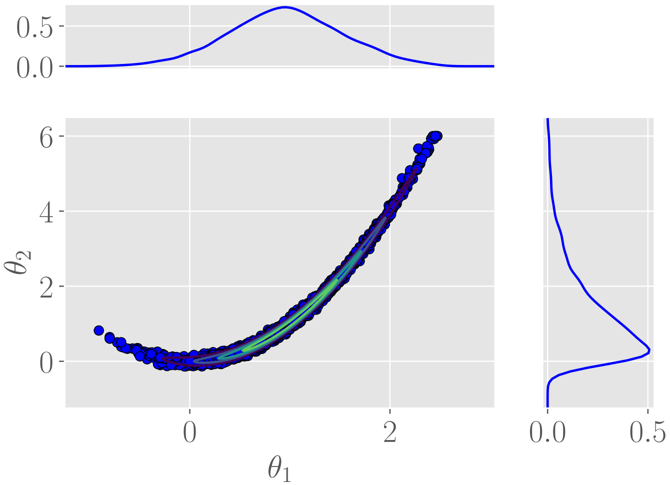
Rosenbrock function

$$\pi_1(\theta_1, \theta_2) = \exp(-12(\theta_2 - \theta_1^2 - 1)^2 + (\theta_1 - 1)^2)$$
$$\pi_2(\theta_1, \theta_2) = \exp(-15(\theta_2 - \theta_1^2)^2 + (\theta_1 - 1)^2)$$

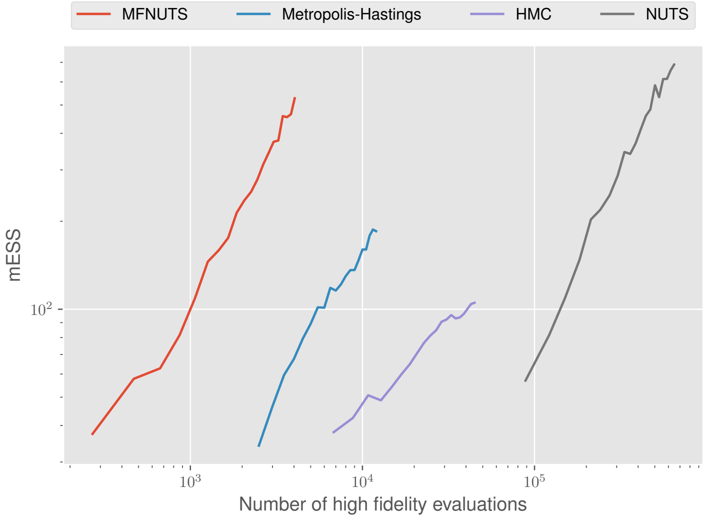


GP surrogate using 50 high-fidelity points and 200 low-fidelity points.

Rosenbrock: Samples



Rosenbrock: mESS vs Computational cost



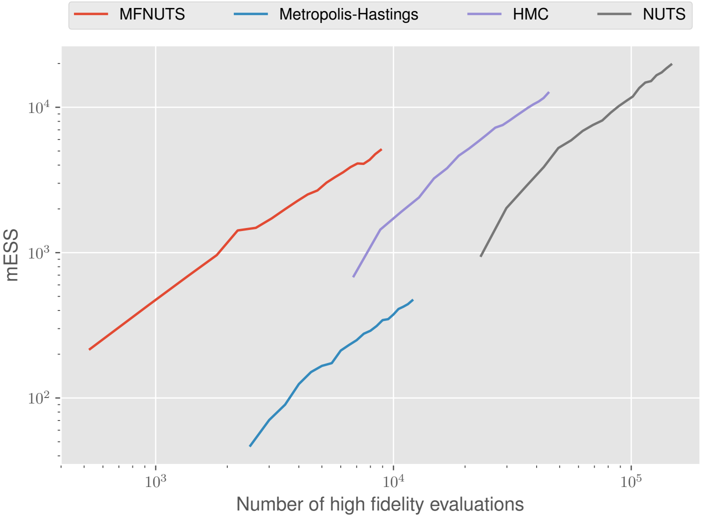
8 dimensional correlated Gaussian

HF function: 8 dimensional correlated gaussian with zero mean

LF function: 8 dimensional gaussian with identity matrix as covariance

GP surrogate using 100 high-fidelity and 500 low-fidelity evaluations

8-d Gaussian: mESS vs Computational cost



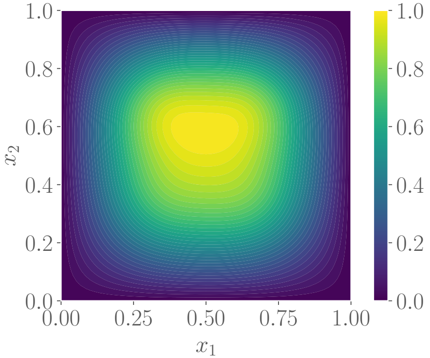
Steady state groundwater flow

- Let us consider a two-dimensional zero-dirichlet steady state groundwater flow problem with source term $S(X)$ and diffusion coefficient $\kappa(X)$

$$\frac{\partial}{\partial X}\left(\kappa(X)\frac{\partial u}{\partial X}\right) = S(X)$$

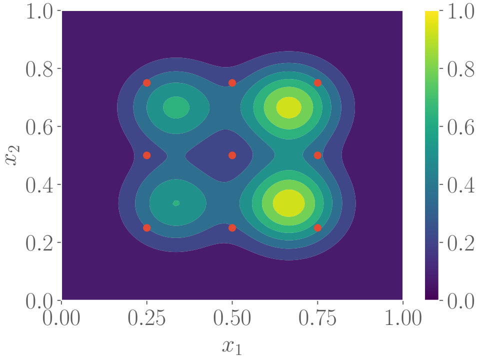
- For this problem, we consider constant $\kappa(X) = 1$ and following source term

$$S(X) = \sum_{i=1}^{N} S_i(X) = \sum_{i=1}^{N} \theta_i \mathcal{N}(\mu_i, \sigma_i^2)$$



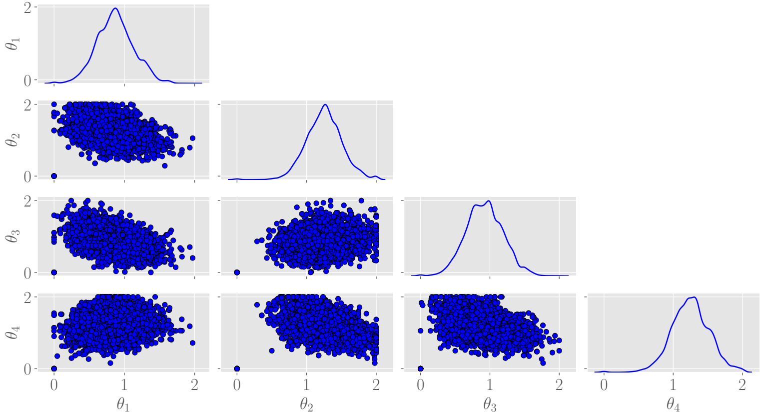
Steady state groundwater flow

- Infer the intensity of source term θ given the observations $u(x)$ at nine probe points marked by orange dots.
- Data is generated by solving the PDE using $\theta = \{0.75, 1.25, 0.8, 1.2\}$ and adding gaussian noise with standard deviation 0.01.
- PDE is solved using open source FEM solver FEniCS[7]
- LF mesh size 8×8, HF mesh size 64×64
- GP surrogate 70 high-fidelity and 450 low-fidelity evaluations



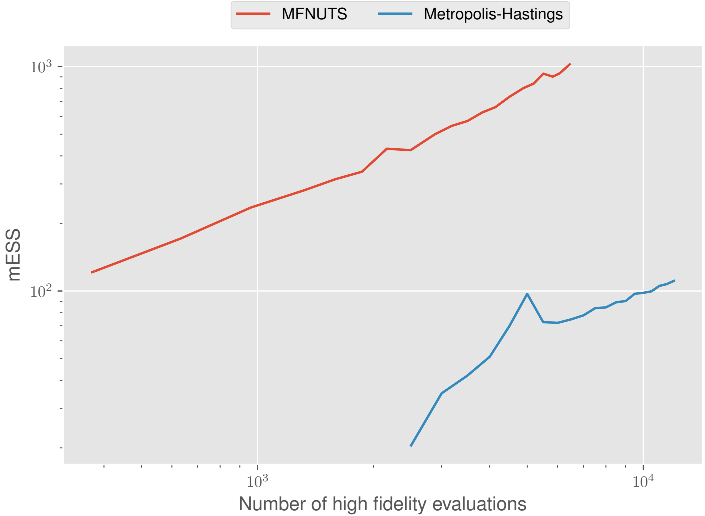
[7] M. S. Alnaes, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes and G. N. Wells. The FEniCS Project Version 1.5, Archive of Numerical Software 3 (2015).

Samples



Mean of samples is $[0.87, 1.25, 0.92, 1.25]$

mESS vs Computational cost



Outline

Conclusion

- MF-NUTS outperforms traditional single fidelity methods.
- We were able to save considerable computational resources by delegating the gradient evaluation to the surrogate.
- The performance of the method depends upon the accuracy of the surrogate.
- One can also use other surrogates.
- Delayed Rejection can be added to further improve the effective sample size.
- Code : `https://github.com/KislayaRavi/MuDaFuGP`

Thank You!

Questions and Feedbacks