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

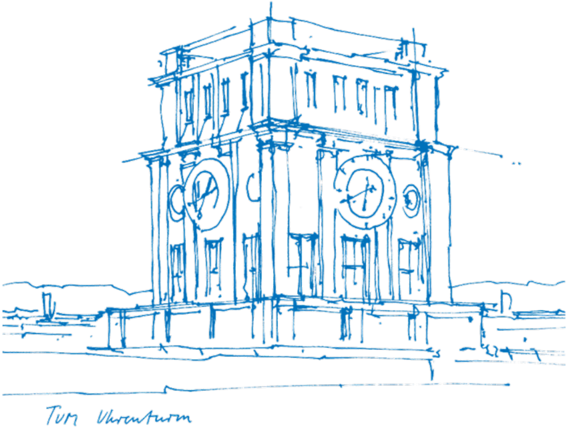
Multi-fidelity No-U-Turn Sampling

MCQMC 2022

Kislaya Ravi

Chair of Scientific Computing
Technical University of Munich

July 19th, 2022



Outline

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 $\implies$ Infeasible for computationally expensive models

MCMC computationaly expensive model

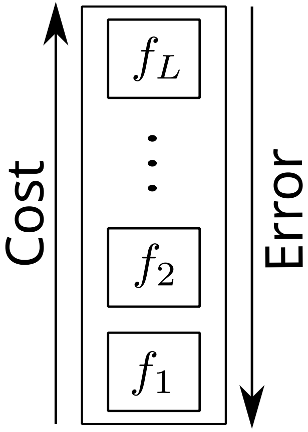
- Sample from a density function which is computationally expensive.
- Becomes challenging for complicated domain/ high-dimensional problems
- Gradient based methods (HMC, NUTS etc.) can help
  - Need Gradient
  - Gradient evaluation is needed at multiple points $\Longrightarrow$ Infeasible for computationally 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, $f_i : \mathbb{R}^d \to \mathbb{R}$ is the i^{th} model

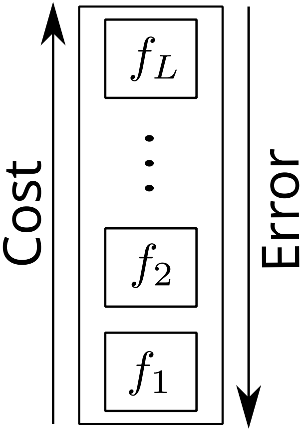


Multi-fidelity

- 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

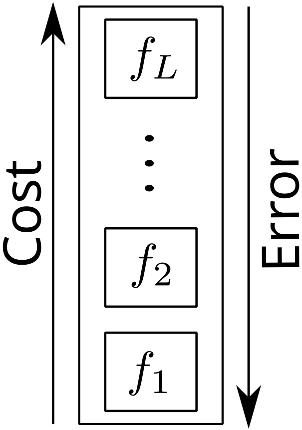


Multi-fidelity

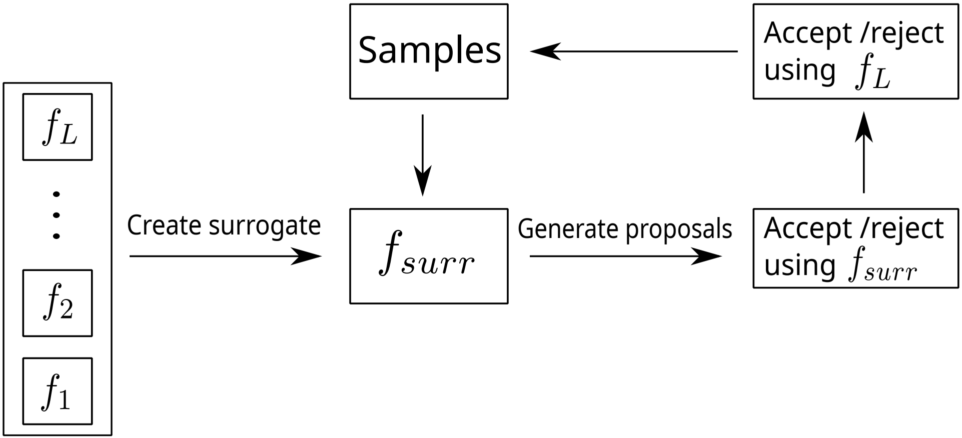
- Supoose we are given ordered set of models as:

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

  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



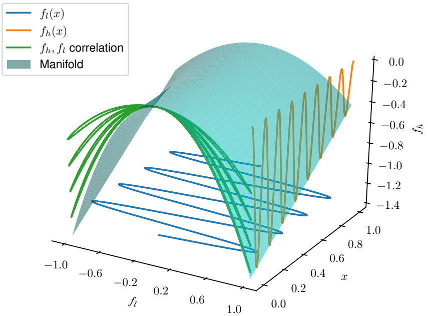
Flowchart


Samples
Accept /reject using f_L
f_L
f_2
f_1
Create surrogate
f_{surr}
Generate proposals
Accept /reject using f_{surr}

Outline

Multi-fidelity implementation[1]

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

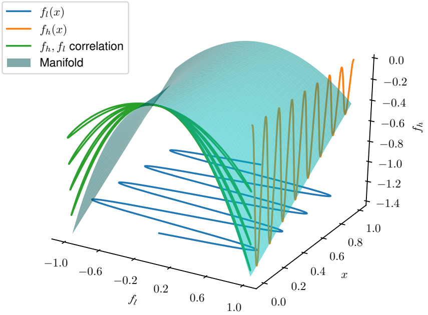


[1] Perdikaris, Paris, et al. "Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473.2198 (2017): 20160751.

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

$$f_h(x) = g(f_l(x), x)$$



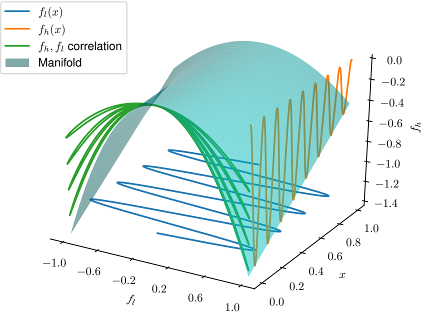
[1] Perdikaris, Paris, et al. "Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473.2198 (2017): 20160751.

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

$$f_h(x) = g(f_l(x), x)$$

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



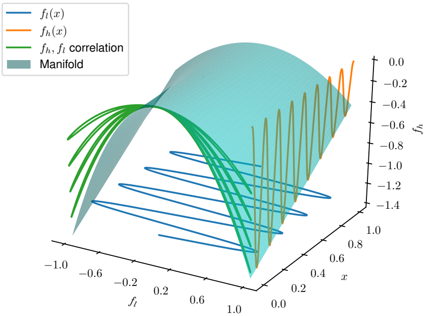
[1] Perdikaris, Paris, et al. "Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473.2198 (2017): 20160751.

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

$$f_h(x) = g(f_l(x), x)$$

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

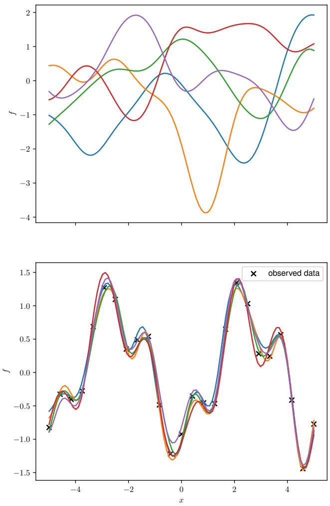


[1] Perdikaris, Paris, et al. "Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473.2198 (2017): 20160751.

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

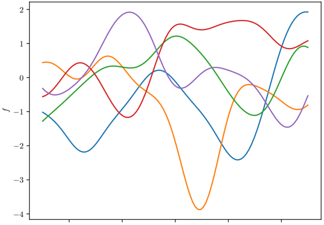
$$f \sim \mathcal{N}(0, K)$$

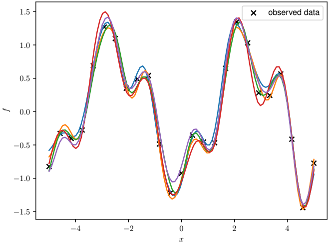
- Prediction at X_* after observing data (X, y) with noise σ^2

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

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

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





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

Gaussian Process [2]

- Gaussian Process is a bayesian model
- Assume prior

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

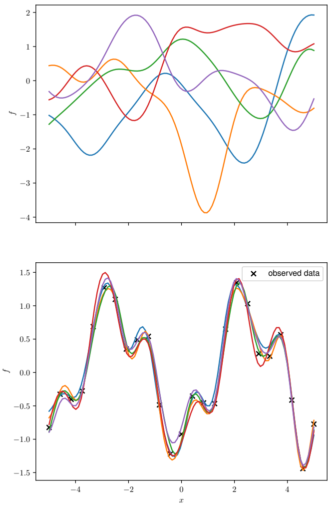
- Prediction at X_* after observing data (X, y) with noise σ^2

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

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

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

- Kernel hyperparameters can be trained by maximizing likelihood



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

Multi-fidelity in GP implementation

- Expand the kernel [1]:

$$K(X, X') = K_\delta(X, X'; \theta_1) + K_\rho(X, X'; \theta_2) K_f(f_l(X), f_l(X'))$$

[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]:

$$K(X, X') = K_\delta(X, X'; \theta_1) + K_\rho(X, X'; \theta_2) K_f(f_l(X), f_l(X'))$$

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

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

[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]:

$$K(X, X') = K_\delta(X, X'; \theta_1) + K_\rho(X, X'; \theta_2) K_f(f_l(X), f_l(X'))$$

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

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

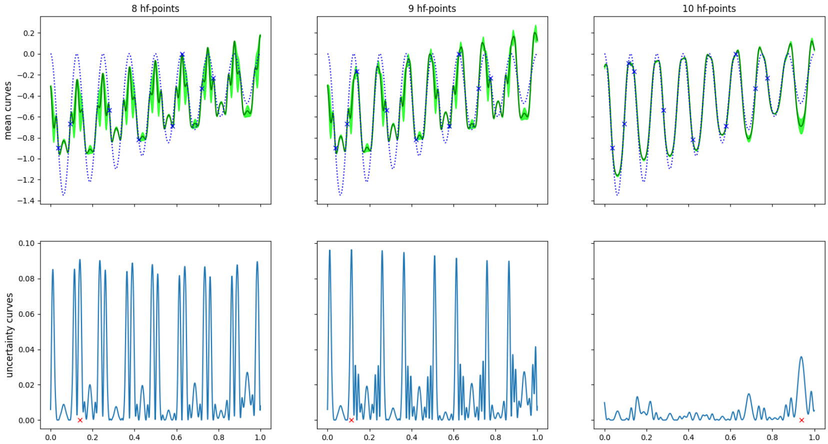
- Adaptively add points where gain of information is maximized:

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

[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.
- Intoduce a momentum term p representing kinetic energy $(K(p))$ and imagine that the negative log of target density represent the potential term $(U(x)) = -log(f(x))$.

[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.
- Intoduce a momentum term p representing kinetic energy $(K(p))$ and imagine that the negative log of target density represent the potential term $(U(x)) = -log(f(x))$.
- Sample from the joint canonical distribution $H(x, p) = K(p) + U(x)$

[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.
- Intoduce a momentum term p representing kinetic energy $(K(p))$ and imagine that the negative log of target density represent the potential term $(U(x)) = -log(f(x))$.
- Sample from the joint canonical distribution $H(x, p) = K(p) + U(x)$
- For the $(i+1)^{th}$ sample:
  - Randomly sample $p \sim \mathcal{N}(0, \mathbb{I}_d)$
  - Solve the Hamiltonian system for some time steps to propose a new point (x', p')
  - Accept/Reject based on Metropolis-Hasting criterion

$$\alpha((x', p'), (x_i, p)) = min\,[1, \exp(H(x', p') - H(x_i, p))]$$

[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.
- Intoduce a momentum term p representing kinetic energy $(K(p))$ and imagine that the negative log of target density represent the potential term $(U(x)) = -log(f(x))$.
- Sample from the joint canonical distribution $H(x, p) = K(p) + U(x)$
- For the $(i+1)^{th}$ sample:
  - Randomly sample $p \sim \mathcal{N}(0, \mathbb{I}_d)$
  - Solve the Hamiltonian system for some time steps to propose a new point (x', p')
  - Accept/Reject based on Metropolis-Hasting criterion

$$\alpha((x', p'), (x_i, p)) = min\left[1, \exp(H(x', p') - H(x_i, p))\right]$$

- 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.
- Intoduce a momentum term p representing kinetic energy $(K(p))$ and imagine that the negative log of target density represent the potential term $(U(x)) = -log(f(x))$.
- Sample from the joint canonical distribution $H(x, p) = K(p) + U(x)$
- For the $(i+1)^{th}$ sample:
  - Randomly sample $p \sim \mathcal{N}(0, \mathbb{I}_d)$
  - Solve the Hamiltonian system for some time steps to propose a new point (x', p')
  - Accept/Reject based on Metropolis-Hasting criterion

$$\alpha((x', p'), (x_i, p)) = min\left[1, \exp(H(x', p') - H(x_i, p))\right]$$

- 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.
- Intoduce a momentum term p representing kinetic energy $(K(p))$ and imagine that the negative log of target density represent the potential term $(U(x)) = -log(f(x))$.
- Sample from the joint canonical distribution $H(x,p) = K(p) + U(x)$
- For the $(i+1)^{th}$ sample:
  - Randomly sample $p \sim \mathcal{N}(0, \mathbb{I}_d)$
  - Solve the Hamiltonian system for some time steps to propose a new point (x', p')
  - Accept/Reject based on Metropolis-Hasting criterion

$$\alpha((x',p'),(x_i,p)) = min\left[1, \exp(H(x',p') - H(x_i,p))\right]$$

- 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:

$$(x - x').p' < 0$$

[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:

$$(x - x').p' < 0$$

- Sample in both directions of the momentum (p and $-p$) 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:

$$(x - x').p' < 0$$

- Sample in both directions of the momentum (p and $-p$) 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:

$$(x - x').p' < 0$$

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

- 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]
- $H(x,p) = K(p) + U(x) = K(p) - log(f_{surr}(x))$
- For the $(i+1)^{th}$ sample:
  - Randomly sample $p \sim \mathcal{N}(0, \mathbb{I}_d)$
  - Generate a proposal using NUTS (x', p')
  - Accept/Reject based using delayed rejection

$$\rho(x', x_i) = min\left[1, \frac{\alpha((x',p'),(x_i,p))f_L(x')}{\alpha((x_i,p),(x',p'))f_L(x_i)}\right]$$

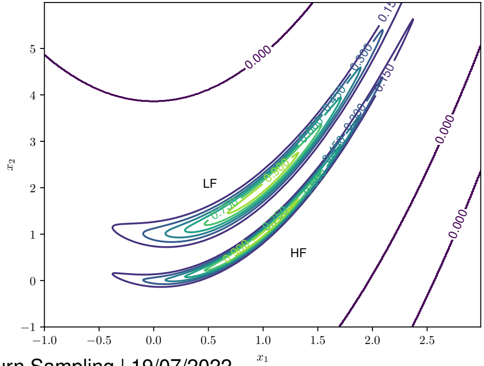
[6] Christen, J. Andrés, and Colin Fox. "Markov chain Monte Carlo using an approximation." Journal of Computational and Graphical statistics 14.4 (2005): 795-810.

Outline

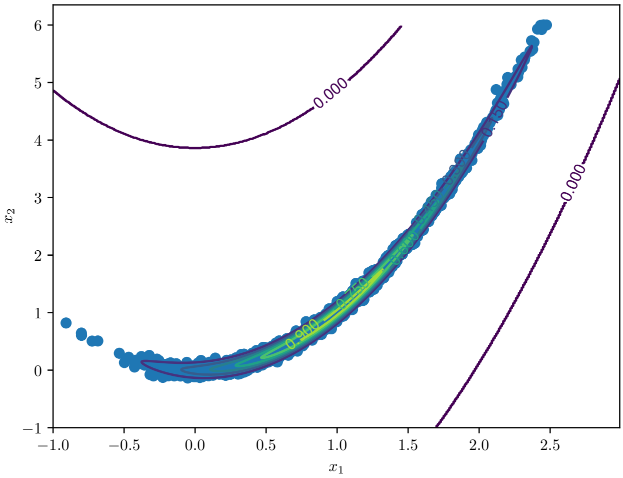
Rosenbrock function

Log of the density function:

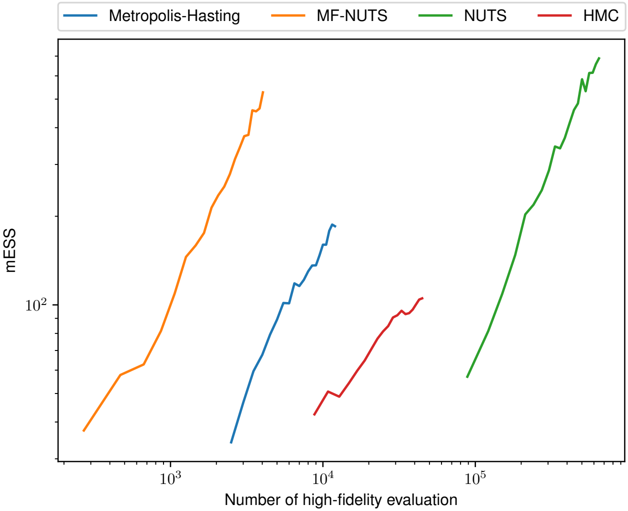
$$p_l(x_1, x_2) \propto f_l(x_1, x_2) = \exp(-12(x_2 - x_1^2 - 1)^2 + (x_1 - 1)^2)$$
$$p_h(x_1, x_2) \propto f_h(x_1, x_2) = \exp(-15(x_2 - x_1^2)^2 + (x_1 - 1)^2)$$



Samples



mESS vs Computational cost

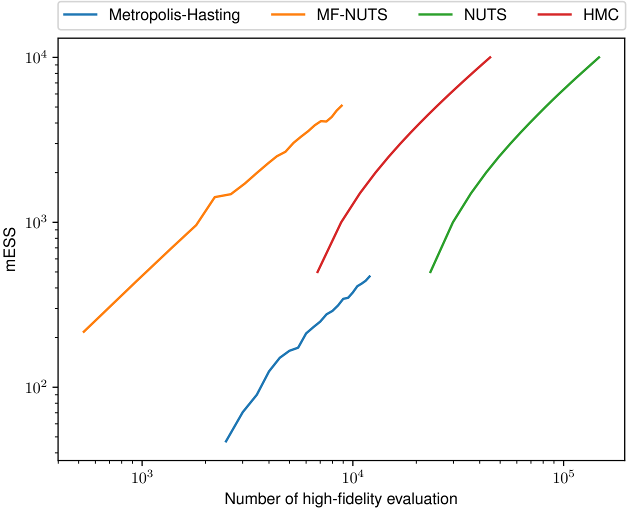


8 dimensional correlated Gaussian

HF function: 8 dimensional correlated gaussian with zero mean

LF function: 8 dimensional gaussian with identity matrix as covariance

mESS vs Computational cost



Outline

Conclusion and Future works

Conclusion

- MF-NUTS outperforms traditional single fidelity methods.
- We were able to save considerable computational resources by delegating the gradient evaluation to the surrogate

[7] Swiler, Laura P., et al. "A survey of constrained Gaussian process regression: Approaches and implementation challenges." Journal of Machine Learning for Modeling and Computing 1.2 (2020).

Conclusion and Future works

Conclusion

- MF-NUTS outperforms traditional single fidelity methods.
- We were able to save considerable computational resources by delegating the gradient evaluation to the surrogate

Future Works

- Create Bayesian Inverse pipeline.
- Test performance for Bayesian Inverse problems.
- Add physics information in gaussian process [7].
- Implement Multi-Output gaussian process for multi-fidelity.

[7] Swiler, Laura P., et al. "A survey of constrained Gaussian process regression: Approaches and implementation challenges." Journal of Machine Learning for Modeling and Computing 1.2 (2020).

Thank You!
Questions and Feedbacks