Stochastic Gradient Hamiltonian Monte Carlo

Sam Adam-Day, Alexander Goodall, Theo Lewy and Fanqi Xu

University of Oxford

Stochastic Gradient Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) already provides a way to sample from a posterior distribution, however it uses all data available to it, which can be computationally expensive for large datasets. This motivated the production of the Stochastic Gradient Hamiltonian Monte Carlo algorithm (SGHMC), which is introduced in [**sghmc**]. It uses batch data to produce noisy estimates of a potential function to, which gives SGHMC dramatic speed-up when compared to HMC. In this paper, Chen et al introduce a Naive SGHMC algorithm, as well as SGHMC itself. They demonstrate links between SGHMC and both Stochastic Gradient Descent with momentum, and Stochastic Gradient Langevin Dynamics. They then run experiments using SGHMC. We reproduced SGHMC, and replicated a number of Chen et al's experiments. The repository for our code is found at https://github.com/sacktock/SGHMC.

Our Reproduction

We reproduced the following experiments from the paper:

- Sampling θ from the posterior with $U(\theta) = -2\theta^2 + \theta^4$ using HMC, Naive SGHMC and SGHMC
- Sampling (θ, r) generated from $U(\theta) = \frac{1}{2}\theta^2$ with HMC, and from $U(\theta) = \frac{1}{2}\theta^2 + \mathcal{N}(0,4)$ as a proxy for SGHMC
- Classifying the MNIST dataset using SGHMC as well as with SGD, SGD with momentum, and SGLD

Our Extensions

We extended this paper in a number of ways:

- We extended the 'No U-Turn Sampler' (NUTS) from [**nuts**] to work with SGHMC to produce our novel algorithm SGNUTS
- We ran experiments on a new dataset of FashionMNIST
- We briefly introduced some Convolutional Neural Networks (CNNs) to see how accurate SGHMC was at classifying CIFAR10
- We attempted to evaluate the noisiness of the data (B in the literature and in what follows) and used this to increase the algorithm's efficiency

The Core Algorithms

The first three are sampling based methods - we sample parameters θ from a model's posterior. We write here the transition step that, upon iterating, gives us $\theta \sim p(\theta|\mathcal{D})$, where $\mathcal{D}$ is all the data available to us. $\tilde{\mathcal{D}}$ is a randomly sampled batch of this data.

Hamiltonian Monte Carlo (HMC)

$$\Delta\theta \leftarrow \epsilon M^{-1} r \qquad \Delta r \leftarrow -\epsilon \nabla U(\theta)$$

where

$$U(\theta) := -\sum_{x \in \mathcal{D}} \log p(x \mid \theta) - \log p(\theta)$$

Naive Stochastic Gradient Hamiltonian Monte Carlo (Naive SGHMC)

$$\Delta\theta \leftarrow \epsilon M^{-1} r \qquad \Delta r \leftarrow -\epsilon \nabla \tilde{U}(\theta)$$

where

$$\tilde{U}(\theta) = -\frac{|\mathcal{D}|}{|\tilde{\mathcal{D}}|} \sum_{x \in \tilde{\mathcal{D}}} \nabla \log p(x \mid \theta) - \nabla \log p(\theta)$$

Stochastic Gradient Hamiltonian Monte Carlo (SGHMC)

$$\Delta\theta \leftarrow \epsilon M^{-1} r \qquad \Delta r \leftarrow -\epsilon \nabla \tilde{U}(\theta) - \epsilon C M^{-1} r + \mathcal{N}(0, 2(C - \hat{B})\epsilon)$$

where

$$\tilde{U}(\theta) = -\frac{|\mathcal{D}|}{|\tilde{\mathcal{D}}|} \sum_{x \in \tilde{\mathcal{D}}} \nabla \log p(x \mid \theta) - \nabla \log p(\theta)$$

and $\hat{B}$ is an estimation of the noise covariance B encapsulated by $\nabla\tilde{U}(\theta) = \nabla U(\theta) + \mathcal{N}(0, 2B\epsilon)$, and C is a hyperparameter

The next two are optimization based methods, which produce θ that are at the mode of the posterior distribution - MAP estimates.

Stochastic Gradient Descent (SGD)

$$\Delta\theta \leftarrow \alpha\Delta\theta - \eta\nabla U(\theta)$$

where α is a momentum hyperparameter. Standard SGD sets $\alpha = 0$

Stochastic Gradient Langevin Dynamics (SGLD)

$$\Delta\theta \leftarrow -\eta\nabla U(\theta) + \mathcal{N}(0, B)$$

where B is the covariance of injected noise.

Reproducing Experiments

Put Fanqi's experiments here. Lorem tempor do enim occaecat in mollit. Ut Lorem adipisicing occaecat nulla cupidatat aute reprehenderit proident. Enim officia ut ex pariatur aute Lorem eu ut duis. Lorem Lorem ut est nostrud aute ullamco. Minim aliquip incididunt occaecat reprehenderit elit irure magna. Do nostrud amet ad ipsum enim sunt. Deserunt velit velit adipisicing exercitation ex fugiat deserunt ullamco eiusmod et consectetur culpa.

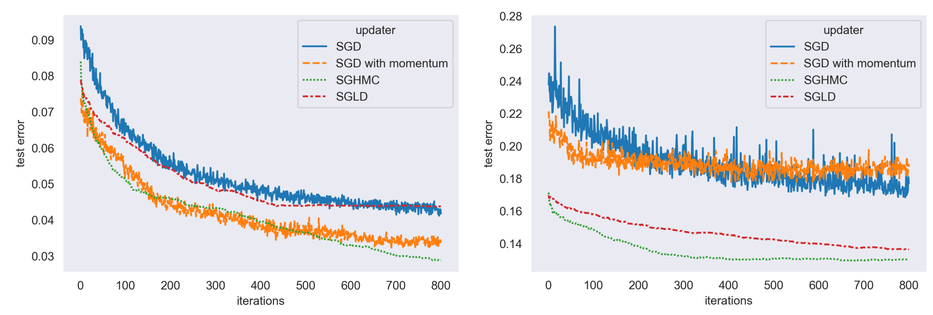

Fig. 1: MNIST (left) and FashionMNIST (right) Classification with SGHMC, SGD, SGD with Momentum, and SGLD

Classifying MNIST

Put Alex's experiments here. Lorem tempor do enim occaecat in mollit. Ut Lorem adipisicing occaecat nulla cupidatat aute reprehenderit proident. Enim officia ut ex pariatur aute Lorem eu ut duis. Lorem Lorem ut est nostrud aute ullamco. Minim aliquip incididunt occaecat reprehenderit elit irure magna. Do nostrud amet ad ipsum enim sunt. Deserunt velit velit adipisicing exercitation ex fugiat deserunt ullamco eiusmod et consectetur culpa.

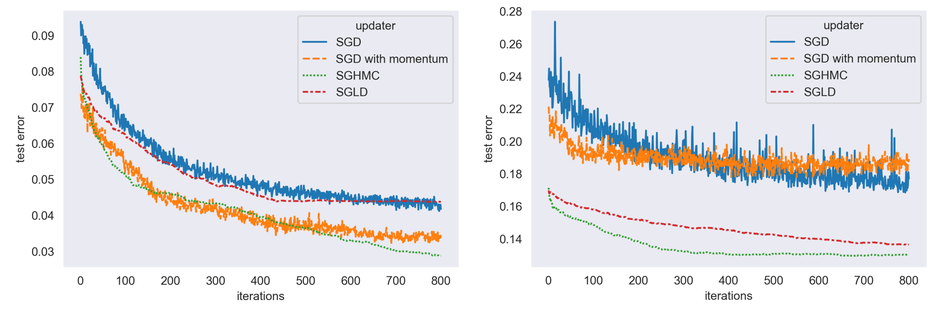

Fig. 2: MNIST (left) and FashionMNIST (right) Classification with SGHMC, SGD, SGD with Momentum, and SGLD

Implementation Details

How things were implemented. Lorem tempor do enim occaecat in mollit. Ut Lorem adipisicing occaecat nulla cupidatat aute reprehenderit proident. Enim officia ut ex pariatur aute Lorem eu ut duis. Lorem Lorem ut est nostrud aute ullamco. Minim aliquip incididunt occaecat reprehenderit elit irure magna. Do nostrud amet ad ipsum enim sunt. Deserunt velit velit adipisicing exercitation ex fugiat deserunt ullamco eiusmod et consectetur culpa. Lorem tempor do enim occaecat in mollit. Ut Lorem adipisicing occaecat nulla cupidatat aute reprehenderit proident. Enim officia ut ex pariatur a

SGNUTS

'Stochastic Gradient No U-Turn Sampler' (SGNUTS) is our novel algorithm based on the 'No U-Turn Sampler' (NUTS) produced in [**nuts**]. NUTS removes the need for the user to pre-set the number of steps HMC performs before taking a sample. We produced SGNUTS to do the same for SGHMC. At its core, it works by repeatedly performing SGHMC steps either forward or backward in time until a 'U-turn' is seen. This is when a further step backwards in time would cause the earliest sample in the trajectory to get closer to the latest sample, or a step forwards in time would cause the latest sample to get closer to the earliest sample. It reached accuracies of 0.94 on MNIST and 0.85 on FashionMNIST, which is similar to SGHMC (accuracies of 0.97 and 0.85 respectively).

Further Research