Stats 225: Bayesian Analysis

More on Markov Chain Monte Carlo

Babak Shahbaba
UC Irvine

Overview

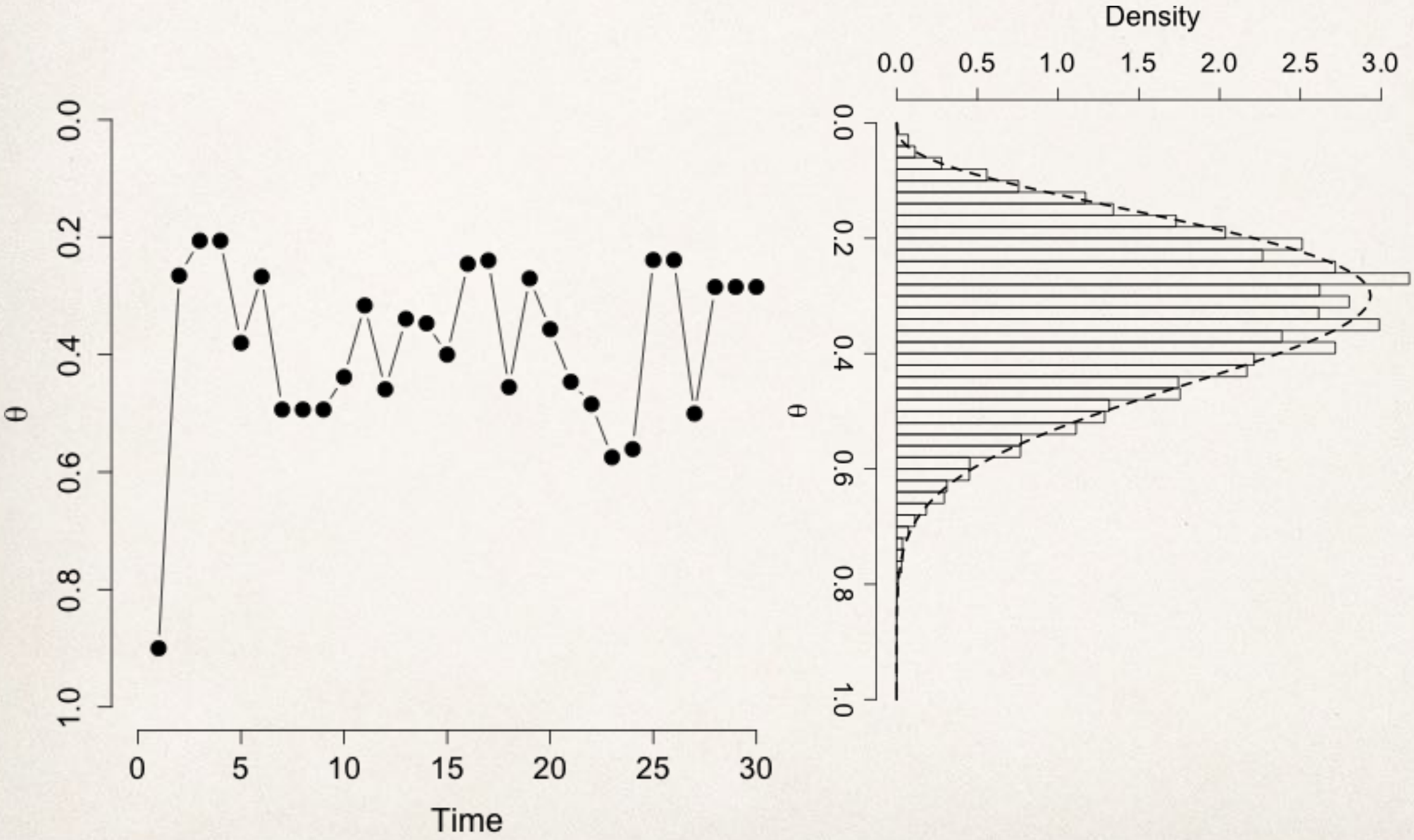
In this lecture, we discuss more advanced computational methods for Bayesian inference.

We first discuss Hamiltonian Monte Carlo (HMC) as an extension of the Metropolis algorithm.

Next, we discuss several variations of HMC to improve its computational efficiency.

Hamiltonian Monte Carlo

Standard Random-Walk Metropolis



Introduction

Simple Metropolis algorithm and Gibbs sampler explore the posterior distribution using a random walk.

While this strategy might work well for low-dimensional distributions, it could become very inefficient (e.g., high autocorrelation, missing isolated modes) for high-dimensional distributions.

Hamiltonian dynamics is used to improve the efficiency of the Metropolis algorithm.

It does this by reducing the random walk behavior through proposing distant states (from the current state) with high probability of acceptance (in theory, with probability 1).

Introduction

In 1953, Metropolis et. al. introduced MCMC to simulate the distribution of states for a system of idealized molecule.

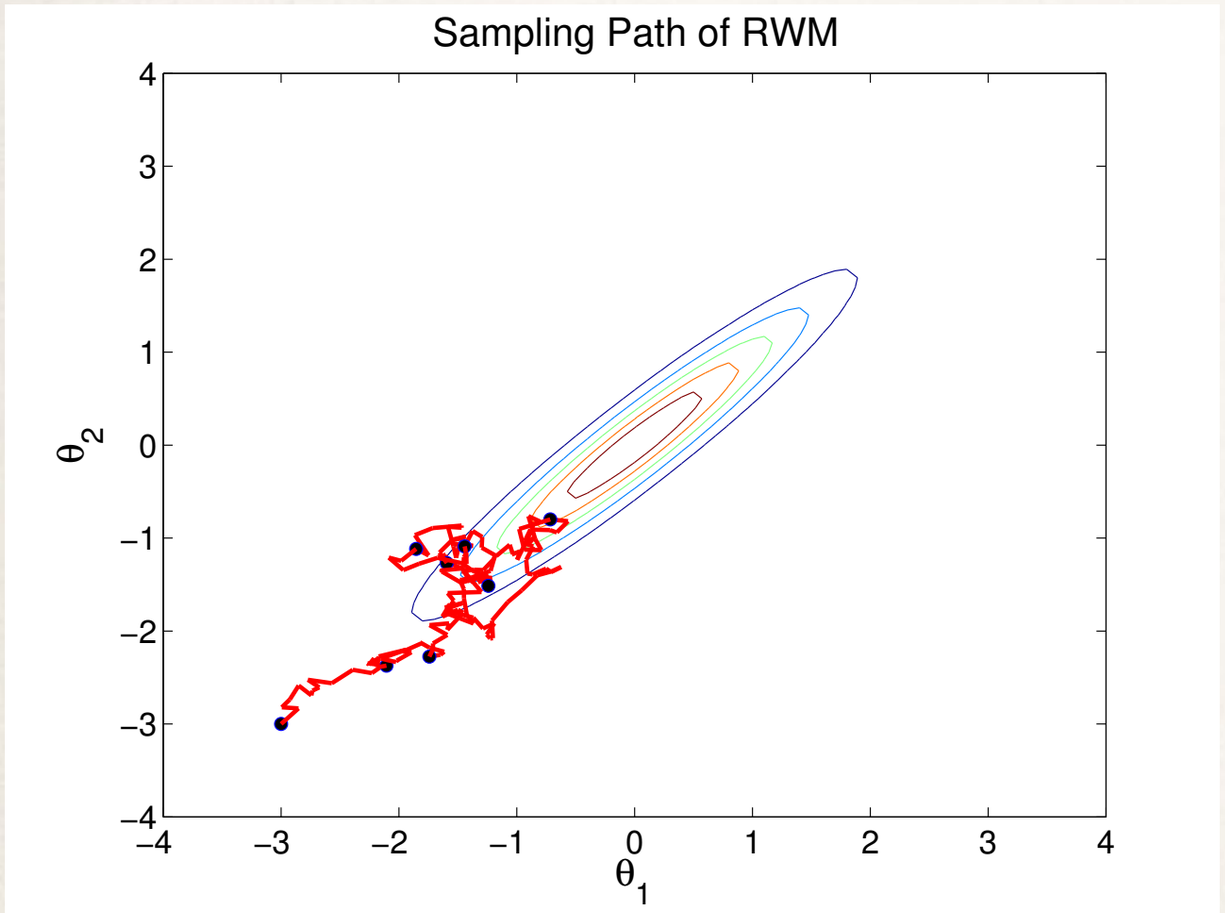
In 1959, Alder and Wainwright introduced a system called *Hamiltonian dynamics* to simulate motion of molecules deterministically based on Newton's law of motion.

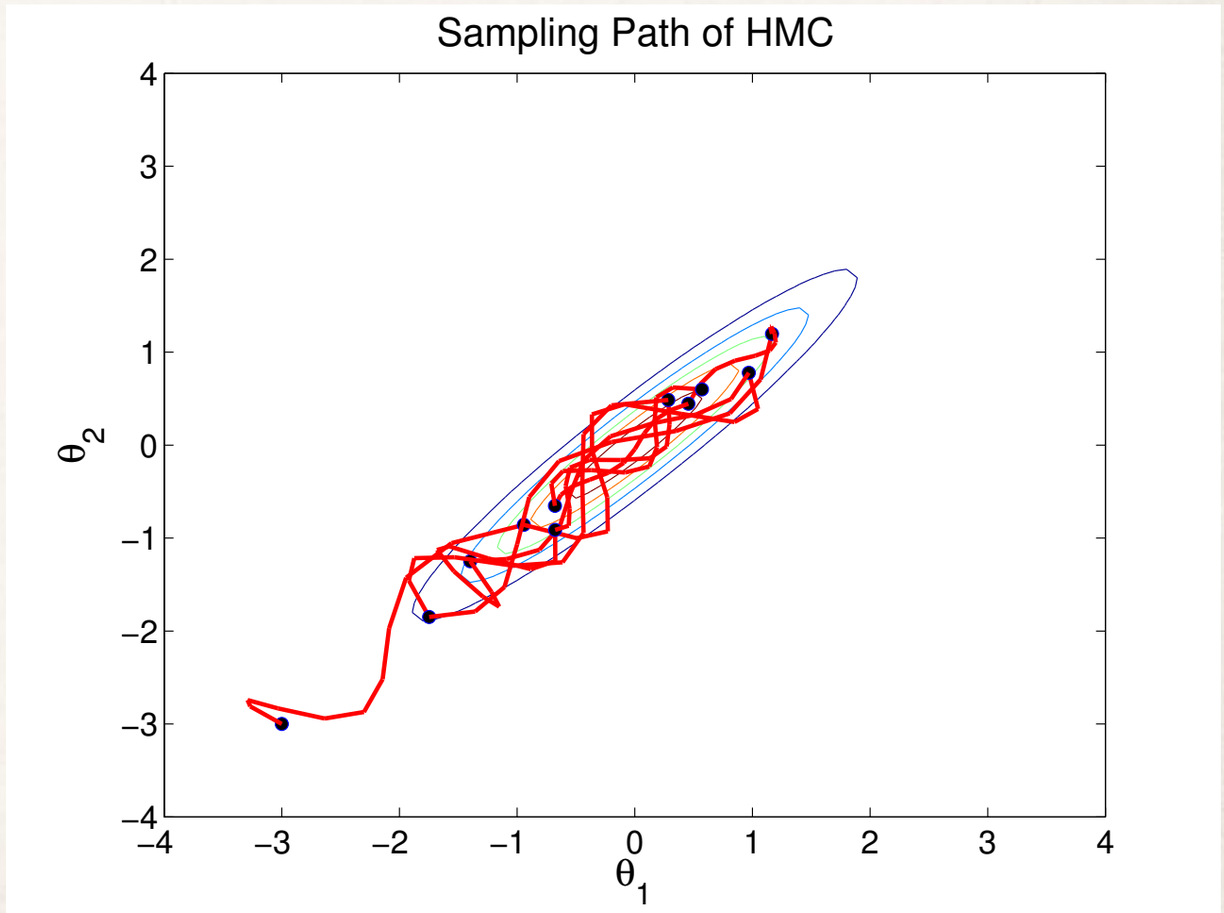
In 1987, Duane et. al. combine the MCMC and the Hamiltonian dynamics. They called their method *Hybrid Monte Carlo*.

The abbreviation HMC has also been used for *Hamiltonian Monte Carlo*.

HMC

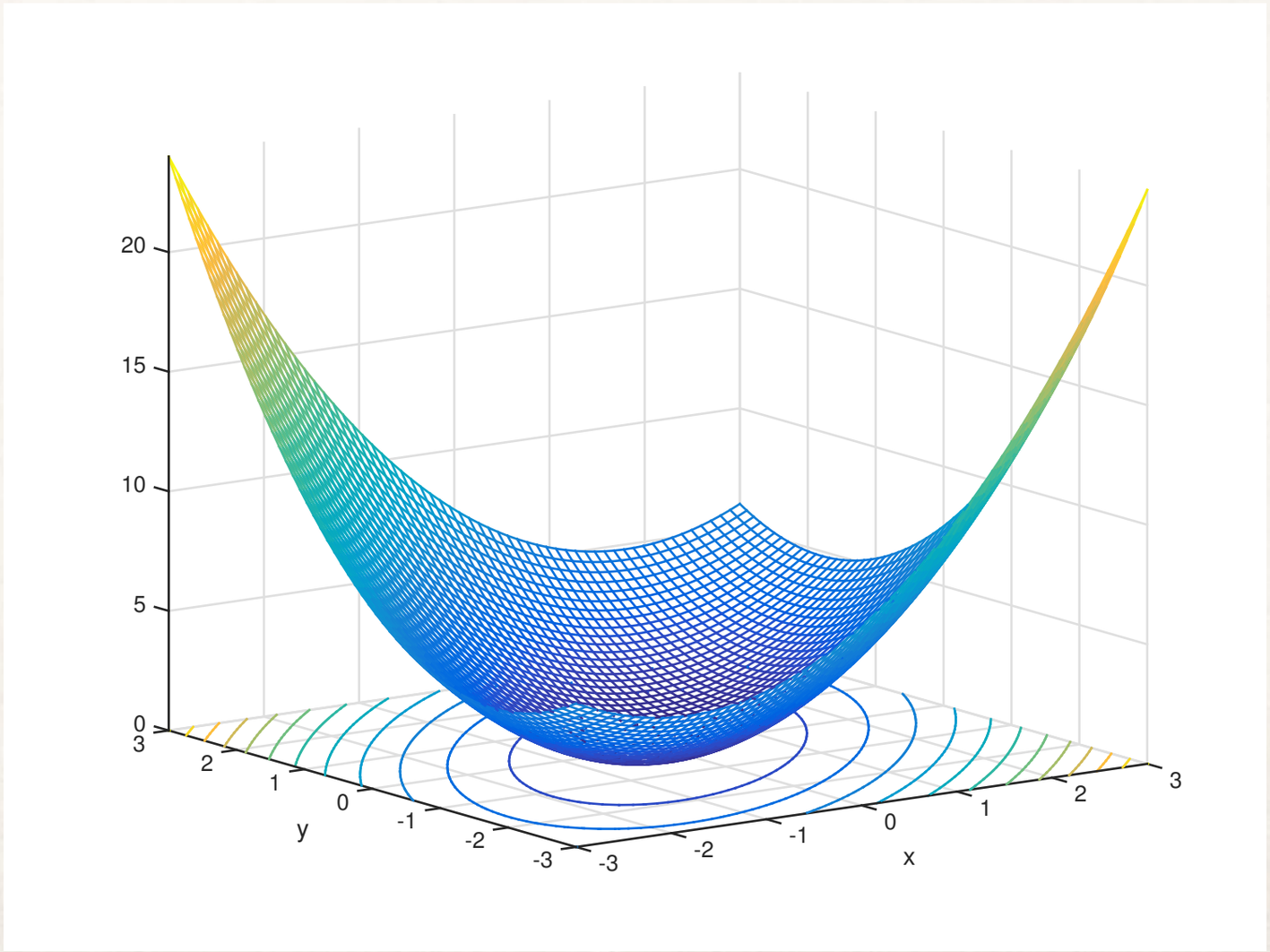
HMC proposes states that are distant from the current state, but nevertheless have a high probability of acceptance.





HMC

The sampler is viewed as a dynamic system moving on a surface defined by the *energy* function$U$: negative log density of the target distribution



Posterior Sampling

For Bayesian inference, posterior distribution is the target distribution

$$U(\theta) = -\sum_{i=1}^{N} \log P(y_i|\theta) - \log P(\theta)$$

We augment the parameter space with fictitious momentum variables

$$K(p) = \frac{1}{2} p^\top M^{-1} p$$

Define the Hamiltonian function $H(\theta, p) = U(\theta) + K(p)$

The joint density of (θ, p) is

$$P(\theta, p) \propto \exp\{-H(\theta, p)\} = \exp\{-U(\theta)\} \cdot \exp\{-K(p)\}$$

The marginal distribution of θ is the posterior distribution.

Posterior Sampling

We can generate a proposal by starting from the current state at time 0 and moving to the state at time t:

$$(\theta, p) = (\theta^{(0)}, p^{(0)}) \longrightarrow (\theta^{(t)}, p^{(t)}) = (\theta^*, p^*)$$

Hamilton's equations determine how θ and $p$change over [fictitious] time

$$\frac{d\theta_j}{dt} = \frac{\partial H}{\partial p_i} = [M^{-1}p]_i$$

$$\frac{dp_j}{dt} = -\frac{\partial H}{\partial \theta_j} = -\frac{\partial U}{\partial \theta_j}$$

Important properties (see http://arxiv.org/pdf/1206.1901.pdf):

- **Reversibility**: the target distribution remain invariant
- **Volume preservation**: the Jacobin determinant is 1
- **Conservation of Hamiltonian**: the acceptance rate is one; θ^* is the next sample if HD is analytically solvable

Posterior Sampling

Numerical integration is employed when analytic solution is not available

$$p_j(t+\epsilon/2) = p_j(t) \; - \; (\epsilon/2)\frac{\partial U}{\partial \theta_j}(\theta(t))$$

$$\theta_j(t+\epsilon) = \theta_j(t) \; + \; \epsilon\frac{\partial K}{\partial p_j}(p(t+\epsilon/2))$$

$$p_j(t+\epsilon) = p_j(t+\epsilon/2) \; - \; (\epsilon/2)\frac{\partial U}{\partial \theta_j}(\theta(t+\epsilon))$$

Important properties:

- **Stability**: numerically stable if ε is appropriately chosen
- **Reversibility and Volume preservation**: still hold
- **Conservation of Hamiltonian**: broken, but can be corrected by MH correction step with acceptance rate:

  $$\alpha = \min[1, \exp(-H(\theta^*, p^*) + H(\theta, p))]$$

HMC Algorithm

Algorithm 1 HMC algorithm

Initialize $\theta^{(0)} = \text{current } \theta$

Sample new momentum $p^{(0)} \sim \mathcal{N}(0, M = I)$

Calculate current $H(\theta^{(0)}, p^{(0)}) = U(\theta^{(0)}) + \frac{1}{2}(p^{(0)})^\top p^{(0)}$

for $\ell = 1$ to L (leapfrog steps) **do**

$$p^{(\ell+\frac{1}{2})} = p^{(\ell)} - \varepsilon/2 \nabla_\theta U(\theta^{(\ell)})$$

$$\theta^{(\ell+1)} = \theta^{(\ell)} + \varepsilon p^{(\ell+\frac{1}{2})}$$

$$p^{(\ell+1)} = p^{(\ell+\frac{1}{2})} - \varepsilon/2 \nabla_\theta U(\theta^{(\ell+1)})$$

end for

Accept or reject according to the Metropolis acceptance probability

A special case: Langevin Monte Carlo

A special case: $L = 1$ and $M = I$

This is called Langevin Monte Carlo,

$$\theta^* = \theta - \frac{\epsilon^2}{2}\nabla_\theta U(\theta) + \epsilon p$$

$$p^* = p - \frac{\epsilon}{2}\nabla_\theta U(\theta) - \frac{\epsilon}{2}\nabla_\theta U(\theta^*)$$

Alternatively, we could ignore the momentum variable $p$and use the following asymmetrical proposal with MH acceptance probability

$$\theta^* \sim N(\theta - \frac{\epsilon^2}{2}\nabla_\theta U(\theta), \epsilon^2 I)$$

Dropping the accept/reject step leads to an approximate Langevin method (see Neal, 1993).

A general case: Riemannian Manifold HMC

Girolami and Calderhead (2011) have introduced a new method, called Riemannian Manifold HMC (RMHMC)

They argue that it is more natural to put the Hamiltonian dynamic on Riemannian manifold of distributions rather than Euclidean space

a They follow Amari (2000) and use the Fisher information matrix, $G(\theta) = E\left[\nabla^2_\theta U(\theta)\right]$, as a metric on the manifold

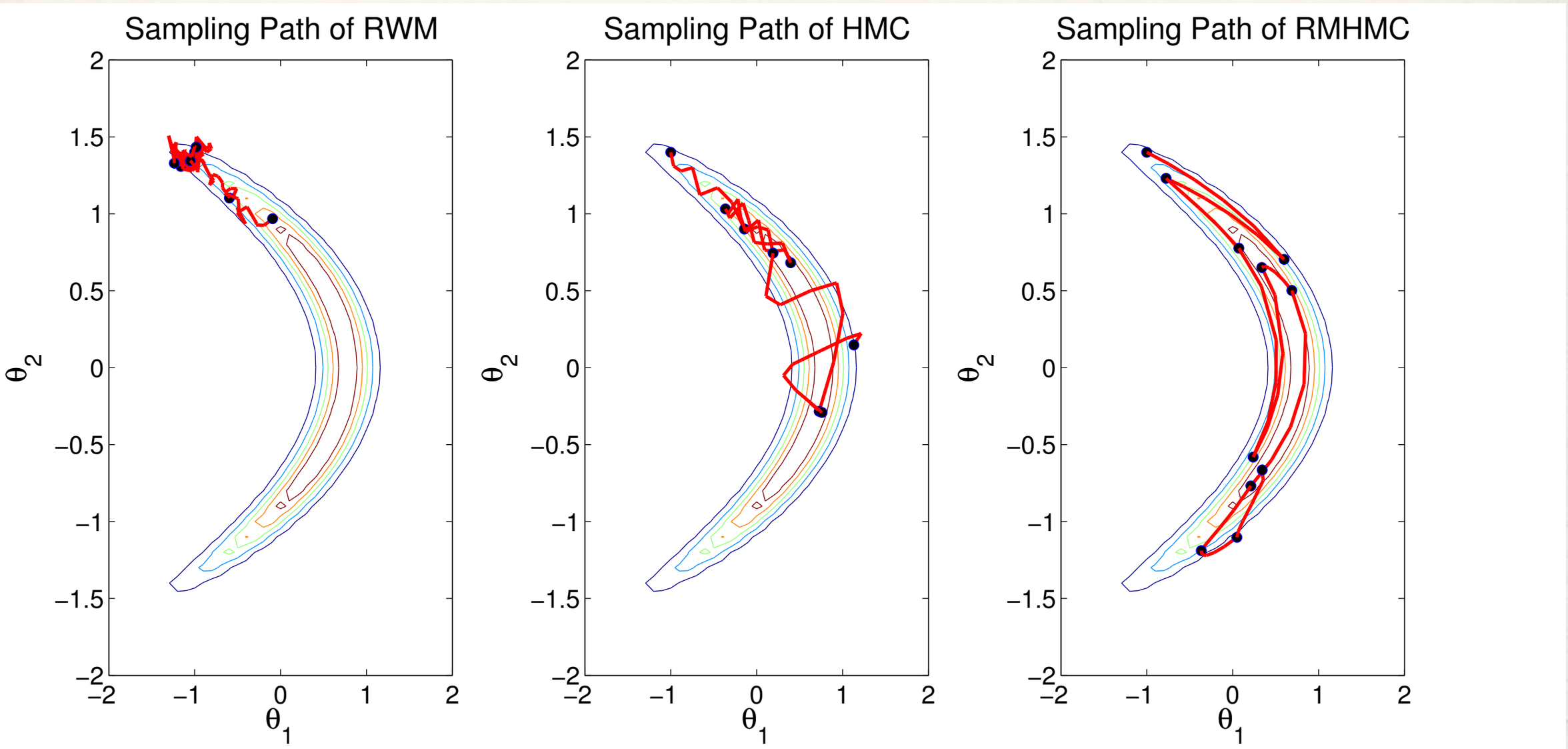
That is, they use position specific mass matrix, $M = G(\theta)$

Example: logistic regression

$$G_{jk}(\beta) = \sum_{i=1}^{N} x_{ij}x_{ik}\frac{\exp(x_{i\beta})}{[1+\exp(x_i\beta)]^2}, \quad j \neq k$$

The resulting dynamics is non-separable so instead of the standard leapfrog method we need to use the *generalized* leapfrog method

A general case: Riemannian Manifold HMC



Scalable HMC

a For high-dimensional problems (big n and/or big d) and complex models, these methods tend to be computationally expensive

To address this issue, we have proposed several variations of HMC:

- Split HMC (S. et al., 2011)
- Lagrangian Monte Carlo (Lan, et al., 2012)
- Spherical HMC (Lan et al., 2013)
- Wormhole HMC (Lan et al., 2013)
- HMC with precomputing strategy (Zhang et al., 2015)
- HMC with surrogate functions (Zhang et al., 2015)

Spherical HMC

Lan, S., Zhou, B., and Shahbaba, B. Spherical Hamiltonian Monte Carlo for Constrained Target Distributions, *ICML* 2014.

Mapping Ball to Sphere

In many cases bounded connected constrained region can be bijectively mapped to *unit ball*

$$\mathbf{B}_0^D(1) := \{\theta \in \mathbb{R}^D : \|\theta\|_2 = \sqrt{\sum_{i=1}^{D} \theta_i^2} \leq 1\}$$

We augment the unit ball to the D-dimensional sphere

$$\mathbf{S}^D := \{\tilde{\theta} \in \mathbb{R}^{D+1} : \|\tilde{\theta}\|_2 = 1\}$$

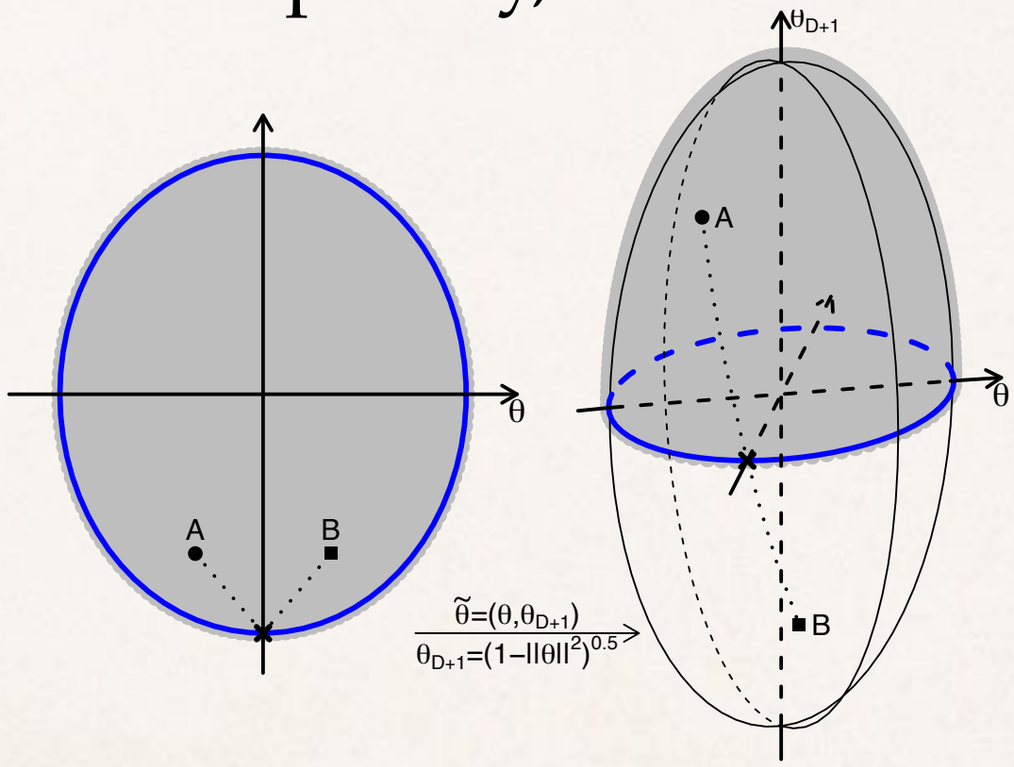
using the following transformation:

$$T_{\mathbf{B}\to\mathbf{S}} : \mathbf{B}_0^D(1) \longrightarrow \mathbf{S}^D, \quad \theta \mapsto \tilde{\theta} = (\theta, \pm\sqrt{1 - \|\theta\|_2^2})$$

HMC on Sphere

Consider an HMC with the Euclidean metric I on the unit ball.

By defining the dynamics on the augmented space (sphere), we handle the constraint implicitly,



The above transformation is equivalent to replacing the metric I with the *canonical spherical metric:* $G_{\mathbf{S}} = I_D + \theta\theta^T/(1 - \|\theta\|_2^2)$.

Norm Constraints

We can apply our method to more general q-norm constraints,

$$\|\beta\|_q = \begin{cases} (\sum_{i=1}^{D} |\beta_i|^q)^{1/q}, & q \in (0, +\infty) \\ \max_{1 \le i \le D} |\beta_i|, & q = +\infty \end{cases}$$

after transforming them to the unit ball.

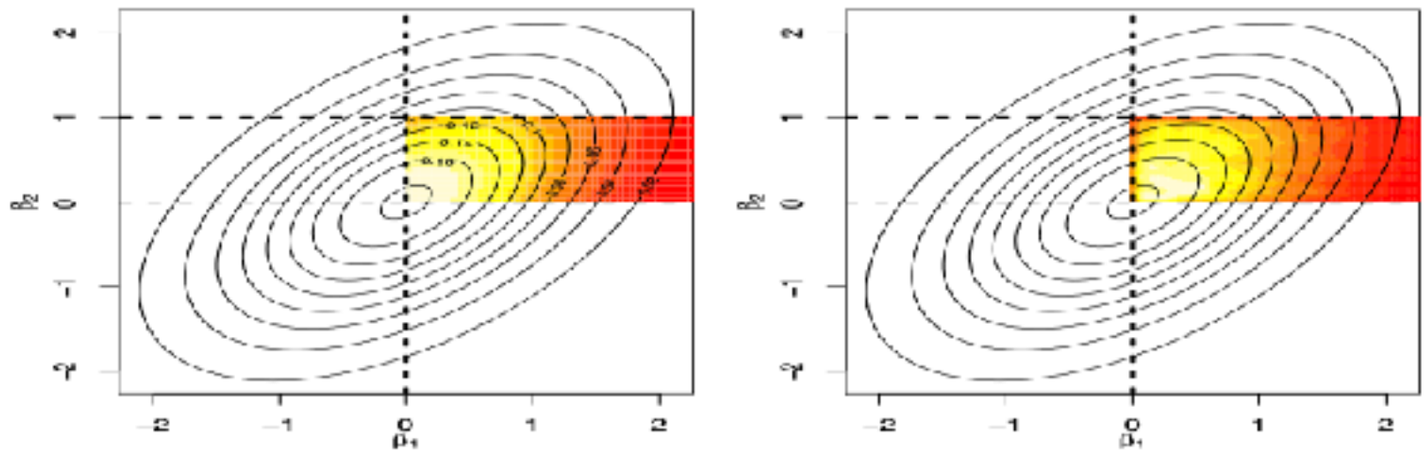
When $q = +\infty$:

$$T_{\mathbf{C} \to \mathbf{B}} : [-1,1]^D \to \mathbf{B}_0^D(1), \quad \beta \mapsto \theta = \beta \frac{\|\beta\|_\infty}{\|\beta\|_2}$$

When $q \in (0, +\infty)$:

$$T_{\mathbf{Q} \to \mathbf{B}} : \mathbf{Q}^D \to \mathbf{B}_0^D(1), \quad \beta_i \mapsto \theta_i = \mathrm{sgn}(\beta_i) |\beta_i|^{q/2}$$

Truncated Normal



Dim	Method	AP	s/Iteration	Min(ESS)	Min(ESS)/s
D=10	RWM	0.64	1.59E-04	15	8.80
	Wall HMC	0.93	5.81E-04	2725	426.79
	Spherical HMC	0.81	9.73E-04	6455	602.78
D=100	RWM	0.72	1.28E-03	1	0.06
	Wall HMC	0.94	1.39E-02	2175	14.23
	Spherical HMC	0.88	1.51E-02	6680	40.12

Split HMC

Shahbaba, B., Lan, S., Johnson, W.O. , Neal, R.M., Split Hamiltonian Monte Carlo, *Statistics and Computing*, 24(3), 339-349.

Splitting the Hamiltonian

We have shown that the technique of "splitting" the Hamiltonian (Leimkuhler and Reich, 2004) can be used to reduce the computational cost of HMC,

$$H(\theta, p) = H_1(\theta, p) + H_2(\theta, p) + \cdots + H_K(\theta, p)$$

The leapfrog method in fact can be regarded as a symmetric splitting of the Hamiltonian

$$H(\theta, p) = U(\theta) + K(p)$$

$$H(\theta, p) = U(\theta)/2 + K(p) + U(\theta)/2$$

Split HMC with a Partial Analytic Solution

Suppose $U(\theta) = U_0(\theta) + U_1(\theta)$

Then, we can split H as

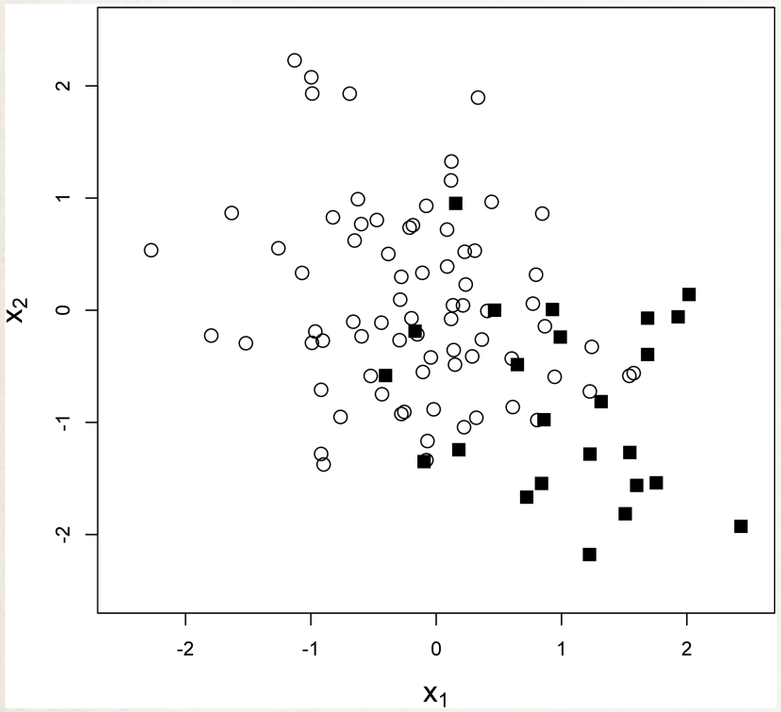
$$H(\theta, p) = U_1(\theta)/2 + [U_0(\theta) + K(p)] + U_1(\theta)/2$$

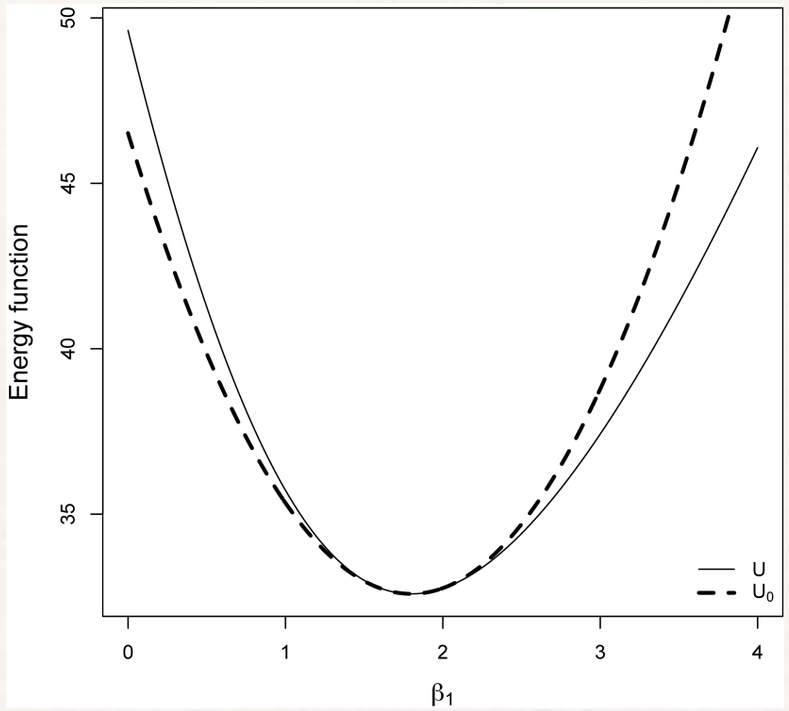
We can define U_0 such that the middle part can be solved analytically to save computation.

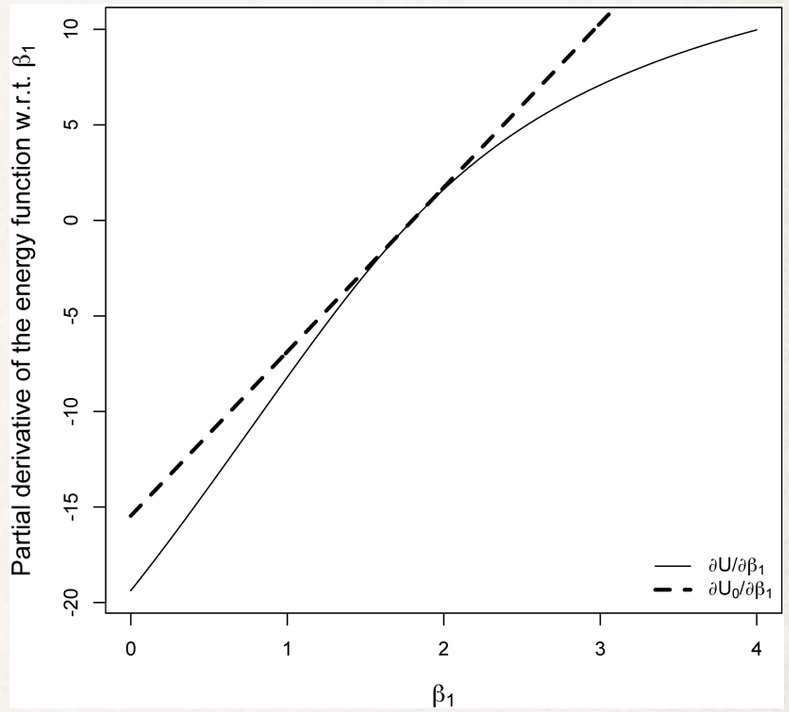
More specifically, we can use the Laplace approximation to define $U_0(\theta)$.

Note that the target distribution remains exact.

Split HMC for Logistic Regression







HMC with Approximated Functions

Zhang, C., Shahbaba, B., Zhao, H. (2017), Precomputing strategy for Hamiltonian Monte Carlo Method based on regularity in parameter space, *Computational Statistics*, 32(1), 253-279.
Zhang, C., Shahbaba, B., Zhao, H. (2017), Hamiltonian Monte Carlo Acceleration Using Surrogate Functions with Random Bases, *Statistics and Computing*, 27, 1473-1490.
Zhang, C., Shahbaba, B., Zhao, H. (2018), Variational Hamiltonian Monte Carlo via Score Matching, *Bayesian Analysis*, 13(2), 485-506.

Subsampling

In recent years, computational methods based on mini-batches of data have been quite successful

- The underlying assumption: there is redundancy in big data
- The overall information can be retrieved from a small subset
- We can approximate functions at low computational cost

Welling and Teh (2011) used this approach (stochastic gradient) for Langevin dynamics using mini-batches of size n from N observations

$$\theta^* = \theta + \frac{\epsilon^2}{2}\left(\nabla_\theta P(\theta) + \frac{N}{n} \sum_{i=1}^{n} \nabla_\theta \log P(x_i | \theta) \right) + \epsilon p$$

They also dropped the accept/reject step

Precomputing Strategies

Finding optimum subsets by exploiting regularity in data space is difficult

Using random subsets could lead to non-ignorable loss of information

Recently, Zhang et. al. (2015) have proposed to switch the focus of approximation from data space to parameter space (http://arxiv.org/abs/1504.01418).

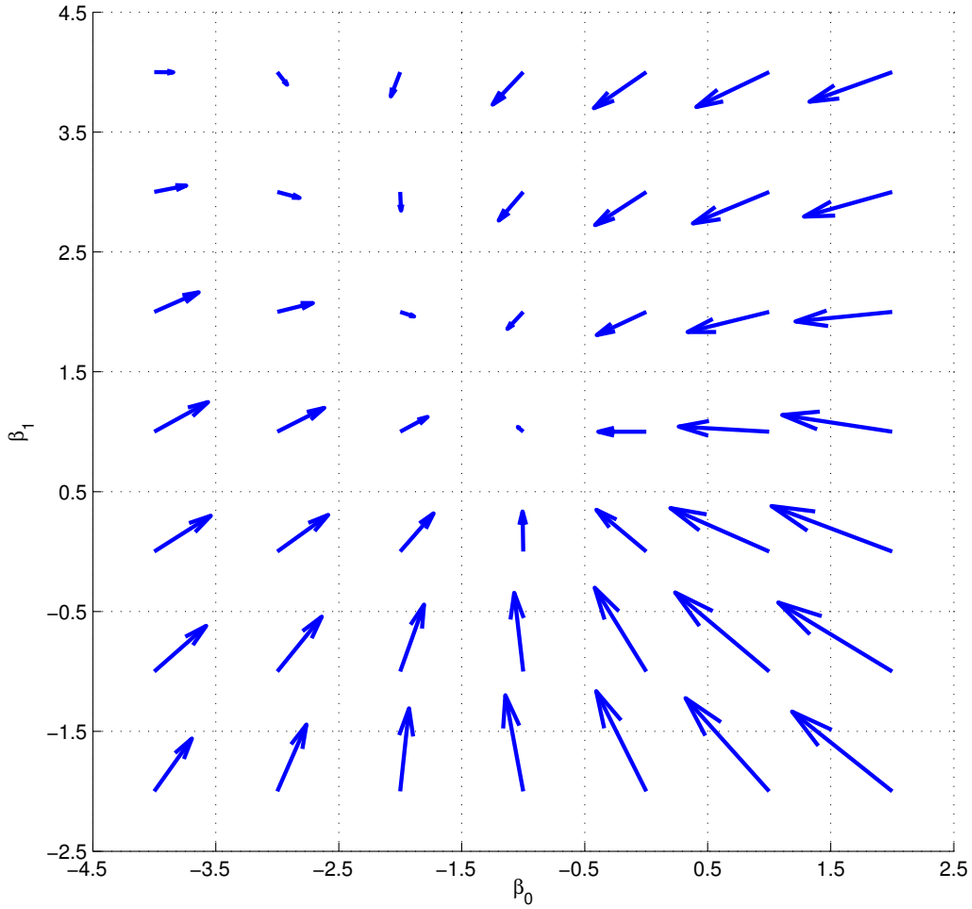
Grid Approximation

Hamilton's equations:

$$\frac{dp_j}{dt} = -\frac{\partial U}{\partial \theta_j}$$

$$\frac{d\theta_j}{dt} = [M^{-1}p]_j$$

Force: $F = -\nabla U$



Piecewise constant approximation

$$\tilde{F}(\theta) = F_{i,j} = F(c_{i,j}), \quad \text{if } \theta \in C_{i,j}$$

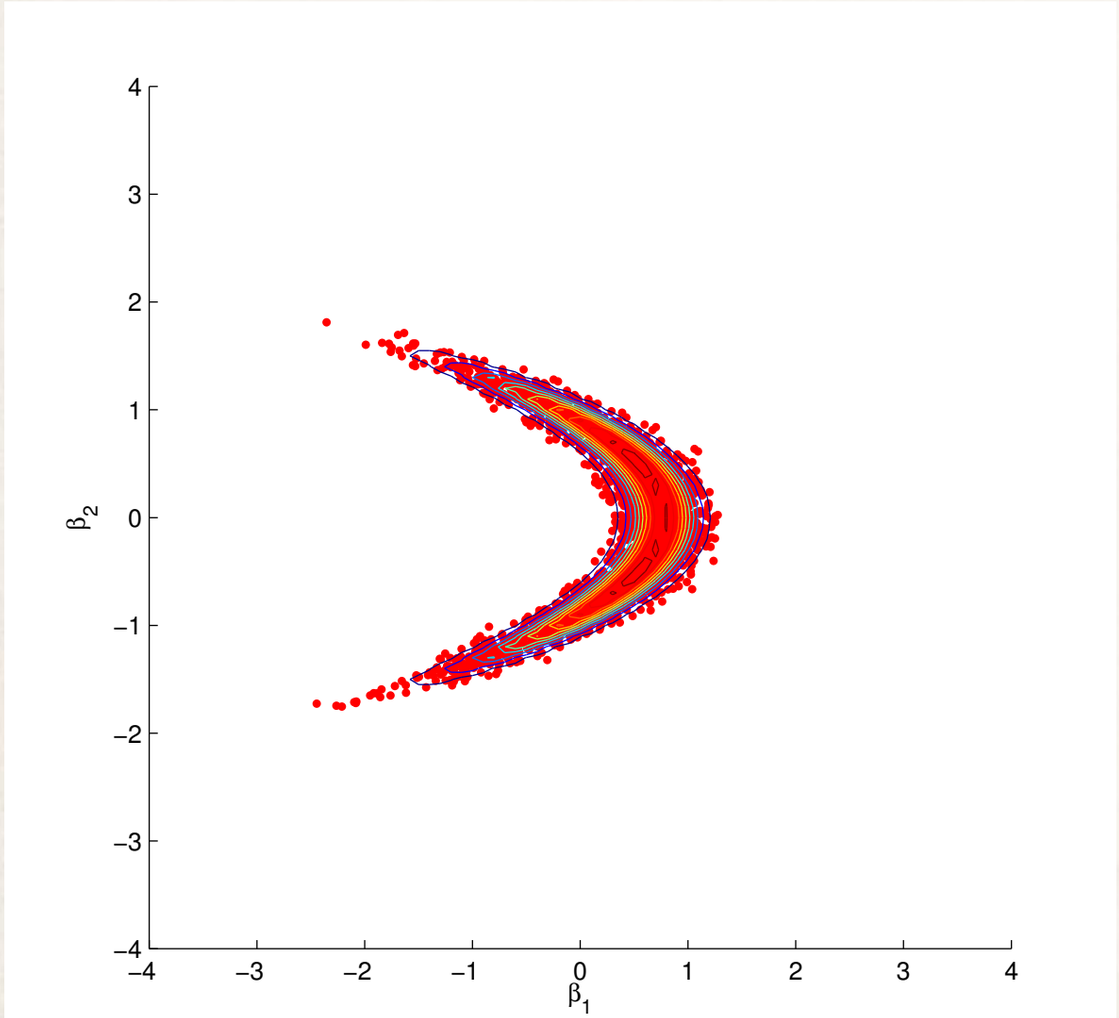
Piecewise linear approximation

$$\tilde{F}(\theta) = F_{i,j} + \nabla F_{i,j} \cdot (q - c_{i,j}), \text{ if } \theta \in C_{i,j}$$

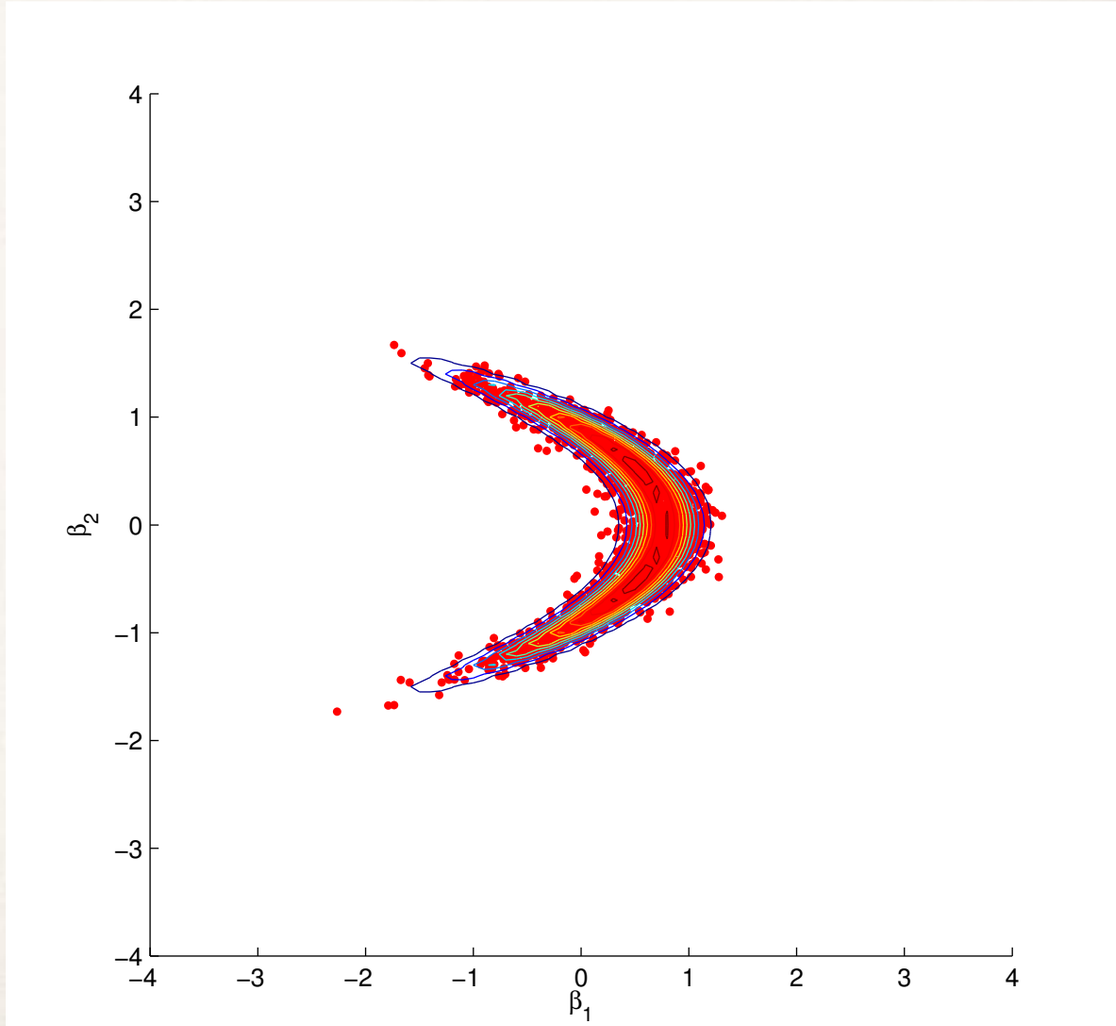
Grid Approximation

Grid HMC is similar to HMC, but it is much faster.

HMC



Grid HMC



Neural Network Surrogate (NNS) HMC

We have instead used a simple generalized additive model, which can be regarded as a shallow neural network,

$$\tilde{U}(\theta) = \sum_{i=1}^{s} v_i g(w_i \cdot \theta + d_i) + d_0$$

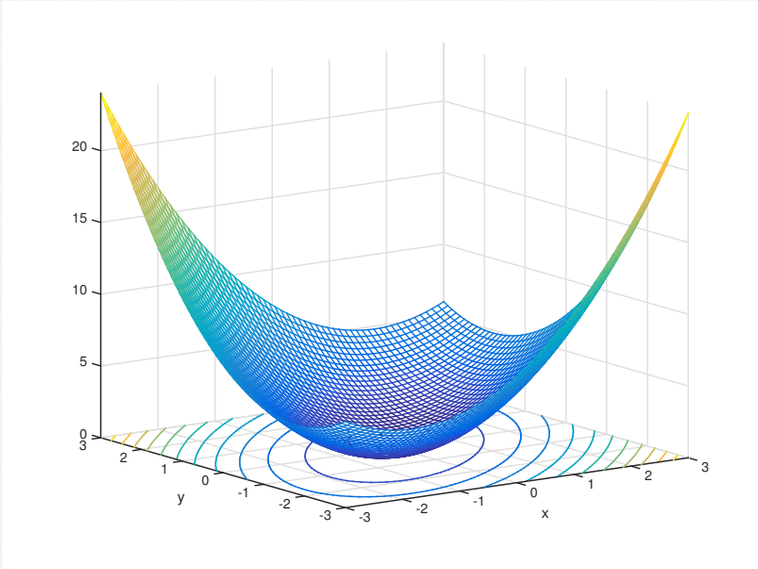
with the softplus function: $g(z) = \log(1 + \exp(z))$

For training, we randomly assign input weights and biases, and then obtain the least-square estimates of the output weights v.

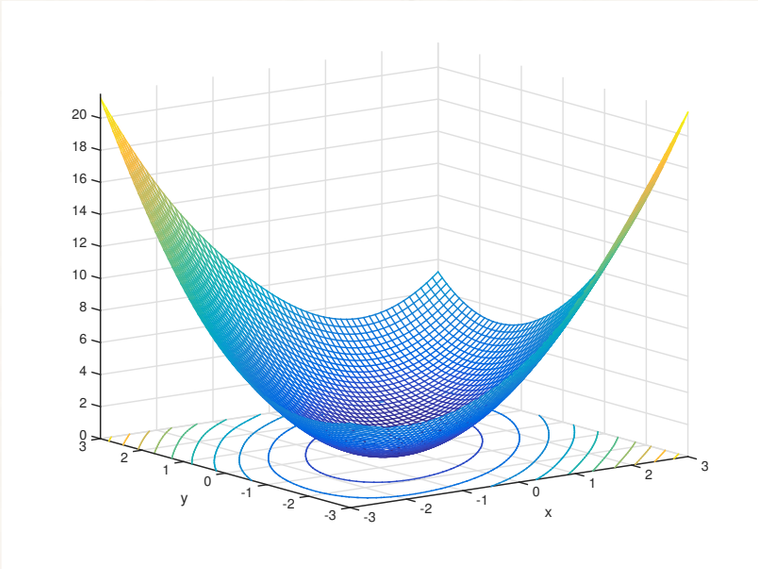
This is known as Extreme Learning Machine (ELM).

Neural Network Surrogate (NNS) HMC

Target Function



NN Approximation



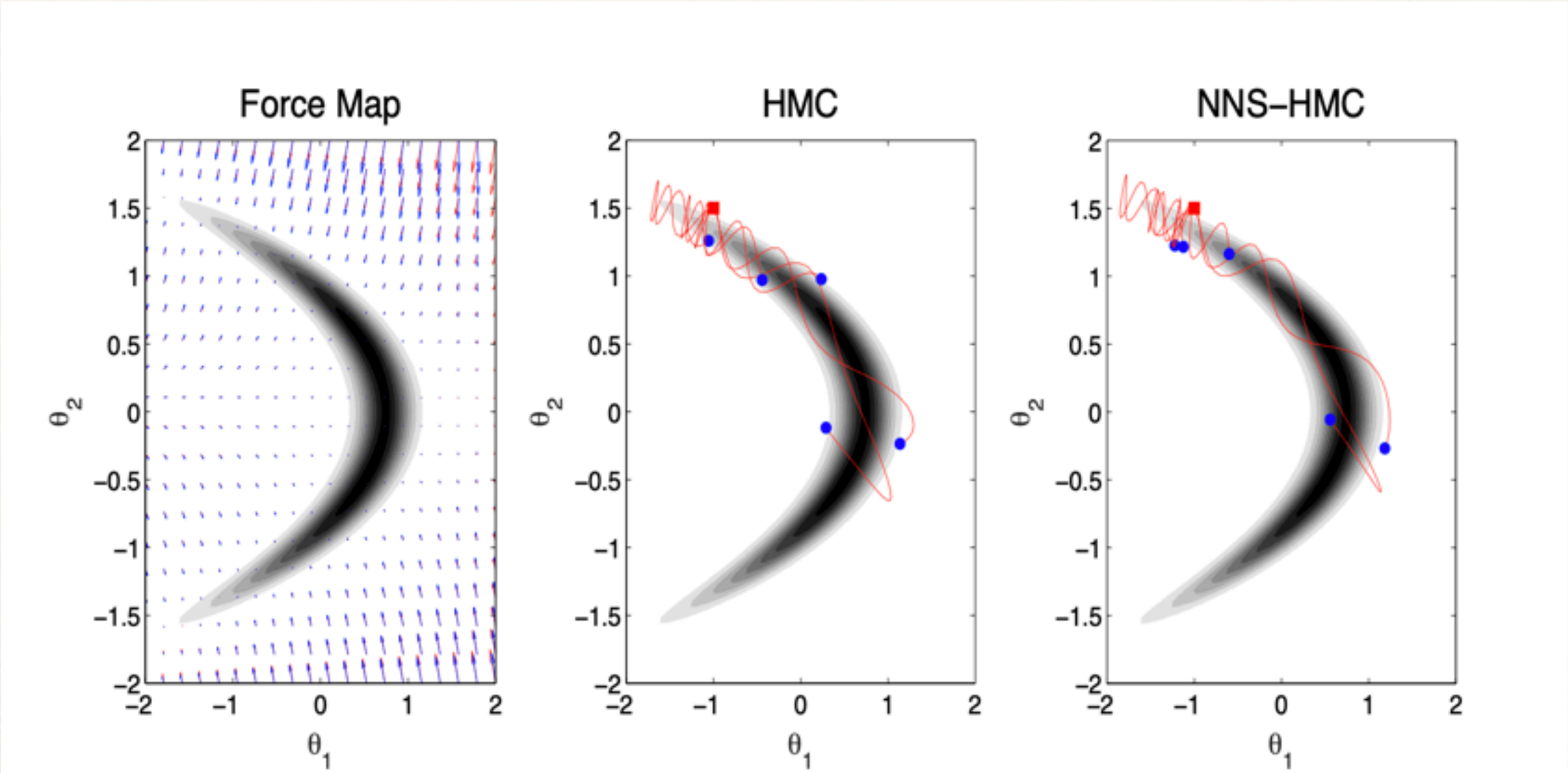
The training process (using pre-convergence samples) and the approximation of functions in the sampling phase can be easily incorporated in HMC

The approximate geometric information (e.g, gradient and Hessian) is obtained by differentiating the neural network directly,

$$\frac{\partial \tilde{U}}{\partial \theta} = \sum_{i=1}^{s} v_i g'(w_i \cdot \theta + d_i) w_i$$

Neural Network Surrogate (NNS) HMC

NNS-HMC follows similar trajectories as HMC, but it is much faster.



Free-form variational Bayes

We can also make inference based on an approximate distribution, similar to variational Bayes, but with a better and more flexible approximation (see for example, de Freitas et al., 2001; Salimans et al., 2015)

For variational Bayes, we typically use a parametrized distribution $q_\eta(\theta)$ to approximate the target posterior $p(\theta \mid Y)$ by minimizing the KL divergence.

Alternatively, we use the approximate distribution based our neural network model

$$Q_v(\theta) \propto \exp(-\tilde{U}(\theta)) = \exp[-\sum_{i=1}^{s} v_i g(w_i \cdot \theta + d_i) + \phi(v)]$$

This is simply a flexible exponential family model

Free-form variational Bayes

To find Q_v, we can follow Hyvarinen (2005) and minimize the score-matching distance

$$\tilde{D}_{SM}(P(\theta|Y)||Q_v(\theta)) = \frac{1}{2}\int Q_v(\theta)\|\nabla_\theta \tilde{U}(\theta) - \nabla_\theta U(\theta)\|^2 \, d\theta$$

For this, we use HMC to generate samples from Q_v

$$\frac{d\theta}{dt} = \frac{\partial \tilde{H}}{\partial p} = M^{-1}p$$

$$\frac{dp}{dt} = -\frac{\partial \tilde{H}}{\partial \theta} = -\nabla_\theta \tilde{U}(\theta)$$

where the modified Hamiltonian is

$$\tilde{H}(\theta, p) = \tilde{U}(\theta) + K(p)$$

Then minimize the regularized empirical distance

$$\hat{v} = \arg\min_v \frac{1}{2}\sum_{n=1}^{t} \|\nabla_\theta \tilde{U}(\theta_n) - \nabla_\theta U(\theta_n)\|^2 + \frac{\lambda}{2}\|v\|^2$$

Example: a beta-binomial model

For illustration, we consider the following beta-binomial model:

$$P(y_j \mid m, K) = \binom{n_j}{y_j} \frac{B(Km + y_j, K(1-m) + n_j - y_j)}{B(Km, K(1-m))}$$