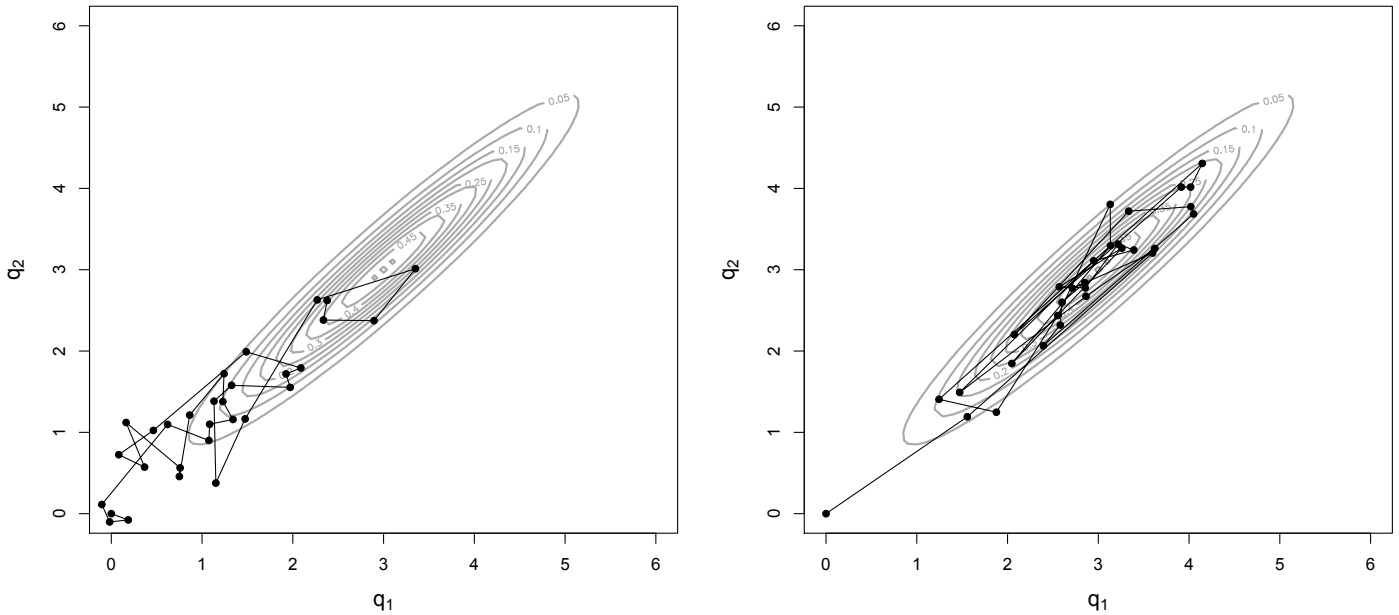
Split Hamiltonian Monte Carlo

Babak Shahbaba, Shiwei Lan, Wesley Johnson, and Radford Neal

July 11, 2012

Objective

- Compare the random walk Metropolis algorithm (left) with Hamiltonian Monte Carlo (right).



- HMC explores the parameter space more efficiently but requires costly gradient evaluations.

- Our objective is to reduce the computational cost of HMC.

The Metropolis algorithm

1. Given our current state $q^{(n)}$, we propose a new state q^*.

2. Calculated the acceptance probability

$$a(q^{(n)}, q^*) = \min(1, \frac{f(q^*)}{f(q^{(n)})})$$

3. With probability $a(q^{(n)}, q^*)$, accept the proposed state q^* as the new state, $q^{(n+1)} = q^*$, or remain at the current state $q^{(n+1)} = q^{(n)}$.

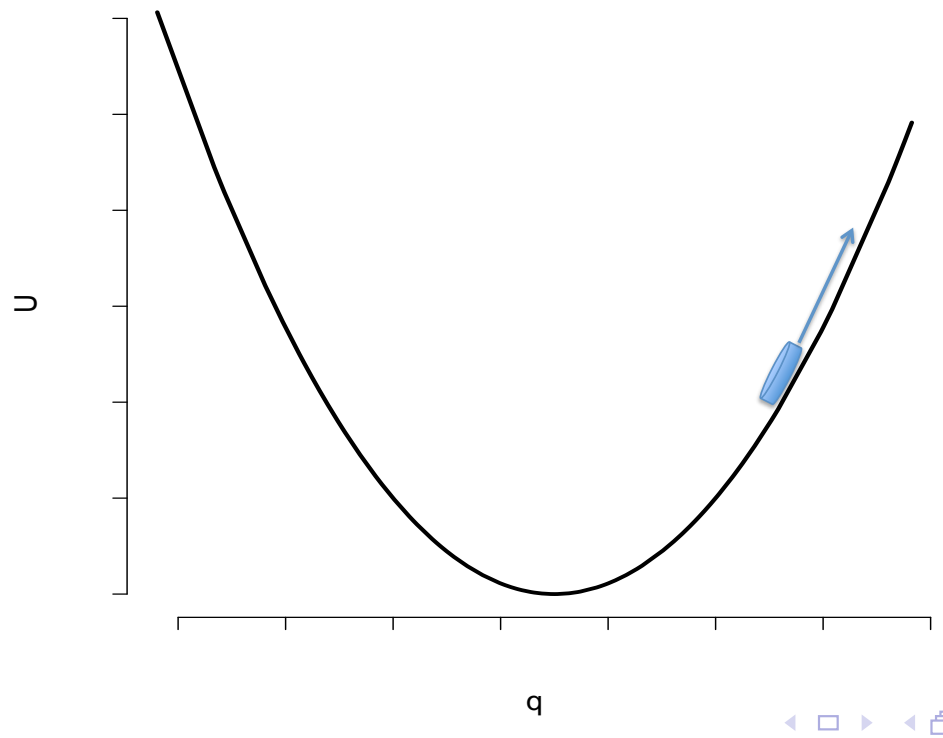
Hamiltonian Monte Carlo

- Hamiltonian Monte Carlo (HMC) reduces the random walk behavior of Metropolis.
- It proposes states that are distant from the current state, but nevertheless have a high probability of acceptance.
- These distant proposals are found by numerically simulating Hamiltonian dynamics for some specified amount of fictitious time.
- Simulation involves costly evaluation of the gradient of the log density.

Physical interpretation of HMC

- Consider a frictionless hockey puck that slides on a surface of varying height.

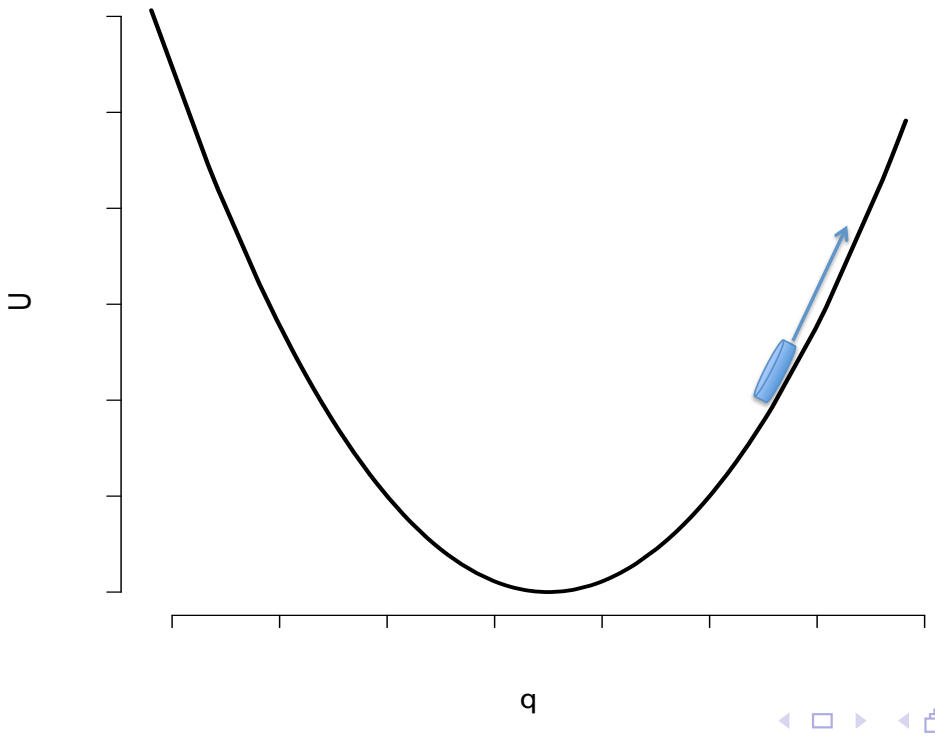
- The state space of this dynamical system consists of its *position* q and its momentum (i.e., mv) denoted as p.



Physical interpretation of HMC

- Based on q, we define the *potential energy*, $U(q)$, such that $U(q)$ is proportional to the height of the surface at position q.

- Based on p, we define the *kinetic energy*, $K(p)$; the kinetic energy is $m|v|^2/2$, so $K(p) = |p|^2/(2m)$.



Hamiltonian's equations

- The dynamic system can be represented by the *Hamiltonian* function:

$$H(q,p) \quad = \quad U(q) + K(p)$$

- *Hamilton's equations* determine how q and p change over time:

$$\frac{dq_j}{dt} \quad = \quad \frac{\partial H}{\partial p_j}$$

$$\frac{dp_j}{dt} \quad = \quad -\frac{\partial H}{\partial q_j}$$

- They define a mapping, T_s, from the state at some time t to the state at time $t + s$.

Sampling

- In Bayesian statistics, q consists of the model parameters,

$$U(q) = -\log(P(q)L(q|D))$$

- We also introduce fictitious momentum variables p.

- Typically, we set $p \sim N(0, M)$ so

$$K(p) = \sum_i p_i^2/2m_i$$

  where M is called the *mass matrix* and is usually set to I.

- The joint density of q and p is

$$\begin{aligned} P(q,p) &= \frac{1}{Z}\exp\Big(-H(q,p)\Big) \\ &= \frac{1}{Z}\exp\Big(-U(q)\Big)\exp\Big(-K(p)\Big) \end{aligned}$$

Properties of HMC

- **Reversibility**; the target distribution remains invariant
- **Conservation of the Hamiltonian**; the acceptance probability is one.
- **Volume preservation**; the determinant of the Jacobian matrix for the mapping is one.
- See Neal (2010) for detailed discussion.

Euler's method

- In practice, we need to approximate these equations by discretizing time, using some small step size ε.

- We can use Euler's method,

$$\begin{aligned} p_j(t+\varepsilon) &= p_j(t) + \varepsilon \frac{dp_i}{dt}(t) = p_j(t) - \varepsilon \frac{\partial U}{\partial q_j}(q(t)) \\ q_j(t+\varepsilon) &= q_j(t) + \varepsilon \frac{dq_j}{dt}(t) = q_j(t) + \varepsilon \frac{\partial K}{\partial p_j}(p(t)) \end{aligned}$$

- However, the approximation error is high.

The leapfrog method

- It is more common to use the *leapfrog* method instead:

$$
\begin{aligned}
p_j(t+\varepsilon/2) &= p_j(t) - (\varepsilon/2)\frac{\partial U}{\partial q_j}(q(t)) \\
q_j(t+\varepsilon) &= q_j(t) + \varepsilon\frac{\partial K}{\partial p_j}(p(t+\varepsilon/2)) \\
p_j(t+\varepsilon) &= p_j(t+\varepsilon/2) - (\varepsilon/2)\frac{\partial U}{\partial q_j}(q(t+\varepsilon))
\end{aligned}
$$

- At the end of the L leapfog steps, we have a propose a new state (q^*, p^*), which is accepted with probability

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

The leapfrog method

Sample initial values for p from $N(0, I)$

for $\ell = 1$ to L **do**

$$p \leftarrow p - (\varepsilon/2)\frac{\partial U}{\partial q}$$

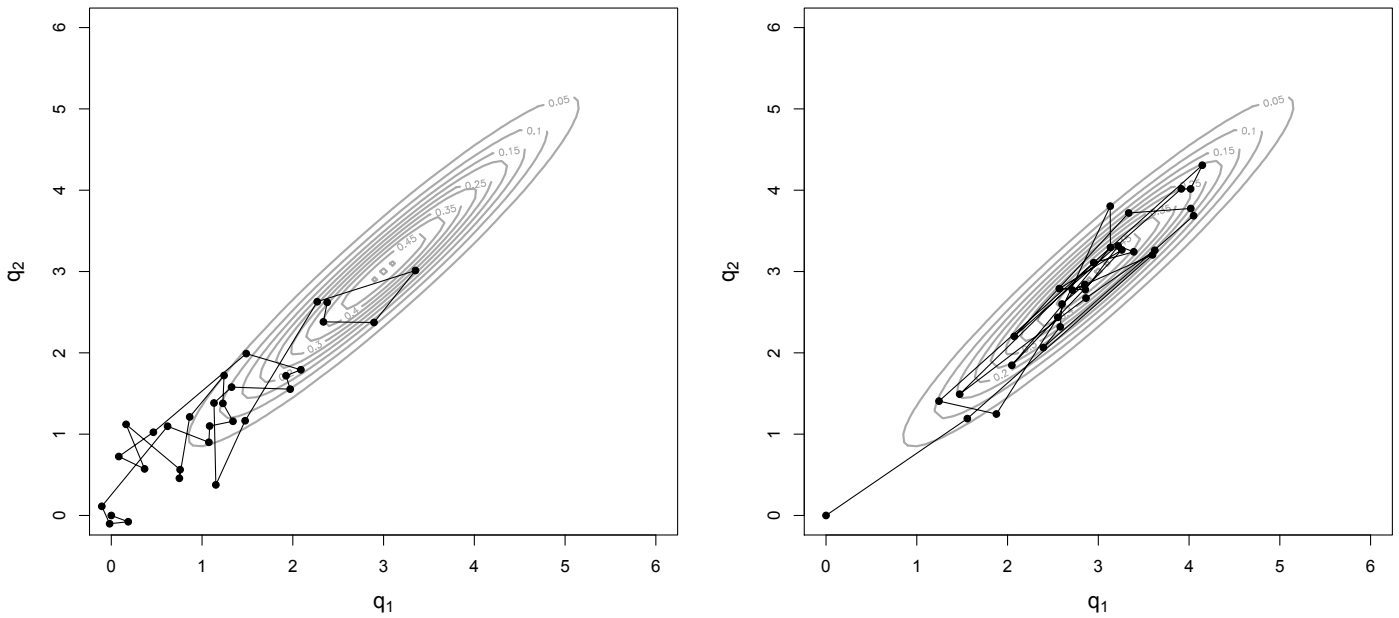
$$q \leftarrow q + \varepsilon p$$

$$p \leftarrow p - (\varepsilon/2)\frac{\partial U}{\partial q}$$

end for

Ilustration

- Consider sampling from a bivariate normal distribution.



- **Left plot**: The first 30 iterations of RWM with 20 updates per iterations. **Right plot**: The first 30 iterations of HMC with 20 leapfrog steps.

Improving HMC

- We still need to find optimum values of ε and L (i.e., optimum trajectory length, $\varepsilon \times L$).
- We also need to choose an appropriate mass matrix, M.
- The simulation requires a costly evaluation of the gradient of the energy function.

NUTS

- Hoffman and Gelman (2011) have tackled the first problem: optimum trajectory length.
- They have proposed a new approach called No-U-Turn Sampler (NUTS).
- NUTS uses a recursive algorithm to build a set of likely candidate points that spans a wide swath of the target distribution, stopping automatically when the trajectory starts to double back and retrace its steps.
- Their method is similar to the doubling procedure proposed by Neal (2003) for slice sampling.

RMHMC

- Girolami and Calderhead (2011) have tackled the second problem: optimum mass matrix.
- They have introduced a new method, called Riemannian Manifold HMC (RMHMC).
- It is more natural to put the Hamiltonian dynamic on Riemannian manifold of distributions rather than Euclidean space.
- They follow Amari (2000) and use the Fisher information matrix, $G(q)$, as a metric on the manifold.
- That is, they use position specific mass matrix, $M = G(q)$

RMHMC

- This way, they reduce autocorrelation by exploiting the geometric properties of the parameter space.
- However, using position specific mass matrix, however, leads to non-separability of Hamiltonian on Riemannian Manifold.
- As a result, RMHMC becomes computationally intensive because simulating from Hamiltonian dynamics involves solving implicit equations, which require additional iterative numerical techniques (e.g., fixed-point iteration).

SGLD

- Further improvement in HMC efficiency can be achieved by using optimization routines within MCMC.
- Recently Welling and Teh (2011) proposed a new approach, called Stochastic Gradient Langevin Dynamics (SGLD), to reduce the computational cost of HMC.
- They combine Langevin dynamics (which can be considered as HMC with only one leapfrog step) with stochastic approximation theory.
- Their approach allows efficient use of mini-batches of data to take advantage of data redundancy.

Split HMC

- We also focus on reducing the computational cost of HMC.
- The computation is mainly dominated by gradient evaluations.
- We show how the technique of "splitting" the Hamiltonian (Leimkuhler and Reich, 2004) can be used to reduce the computational cost of HMC,

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

- The leapfrog method in fact can be regarded as a symmetric splitting of the Hamiltonian $H(q,p) = U(q) + K(p)$ as

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

Split HMC with a partial analytic solution

- Suppose $U(q) = U_0(q) + U_1(q)$.
- Then, we can split H as

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

- Suppose the middle part can be handled analytically.
- Then its simulation introduces no error.
- We should be able to use a larger step size and fewer steps.

Split HMC with a partial analytic solution

- We approximate $U(q)$ by $U_0(q)$.
- We set $U_1(q) = U(q) - U_0(q)$, the error in this approximation.
- Specifically, we set $U_0(q)$ to the energy function for $N(\hat{q}, \mathcal{J}^{-1}(\hat{q}))$.
- Here, $\hat{q}$ is the MAP estimate, and $\mathcal{J}(\hat{q})$ is the Hessian matrix of U at $\hat{q}$.

Split HMC with a partial analytic solution

- We set $H_2(q,p) = U_0(q) + K(p)$.
- The corresponding Hamilton's equations will be a system of first-order linear differential equations that can be handled analytically.
- We set

$$A = \begin{bmatrix} 0 & I \\ -\mathcal{J}(\hat{q}) & 0 \end{bmatrix}$$

- Then, we diagonalize $A = \Gamma D \Gamma^{-1}$.

Algorithm 1

$R \leftarrow \Gamma e^{D\varepsilon}\Gamma^{-1}$

Sample initial values for p from $N(0, I)$

for $\ell = 1$ to L **do**

$p \leftarrow p - (\varepsilon/2)\dfrac{\partial U_1}{\partial q}$

$q^* \leftarrow q - \hat{q}$

$X_0 \leftarrow (q^*, p)$

$(q^*, p) \leftarrow RX_0$

$q \leftarrow q^* + \hat{q}$

$p \leftarrow p - (\varepsilon/2)\dfrac{\partial U_1}{\partial q}$

end for

Split HMC by splitting the data

- Next, suppose $H_2(q, p)$ cannot be handled analytically.
- However, suppose that the computational cost for $U_0(q)$ is still substantially lower than for $U(q)$.
- In these situations, we can use the following split:

$$\begin{aligned} H(q,p) &= U_1(q)/2 + \\ &\quad \sum_{m=1}^{M} [U_0(p)/2M + K(p)/M + U_0(p)/2M] + \\ &\quad U_1(q)/2 \end{aligned}$$

Split HMC by splitting the data

- For example, suppose our statistical analysis involves a large data set with many observations.
- We can construct $U_0(q)$ based on a small part of the observed data, R_0.
- We use the remaining observations, R_1, to construct $U_1(q)$.

$$
\begin{aligned}
U(\theta) &= U_0(\theta) + U_1(\theta) \\
U_0(\theta) &= -\log[P(\theta)] - \sum_{i \in R_0} \log[P(y_i|\theta)] \\
U_1(\theta) &= -\sum_{i' \in R_1} \log[P(y_{i'}|\theta)]
\end{aligned}
$$

Algorithm 2

Sample initial values for p from $N(0, I)$

for $\ell = 1$ to L **do**

$$p \leftarrow p - (\varepsilon/2)\frac{\partial U_1}{\partial q}$$

for $m = 1$ to M **do**

$$p \leftarrow p - (\varepsilon/2M)\frac{\partial U_0}{\partial q}$$

$$q \leftarrow q + (\varepsilon/M)p$$

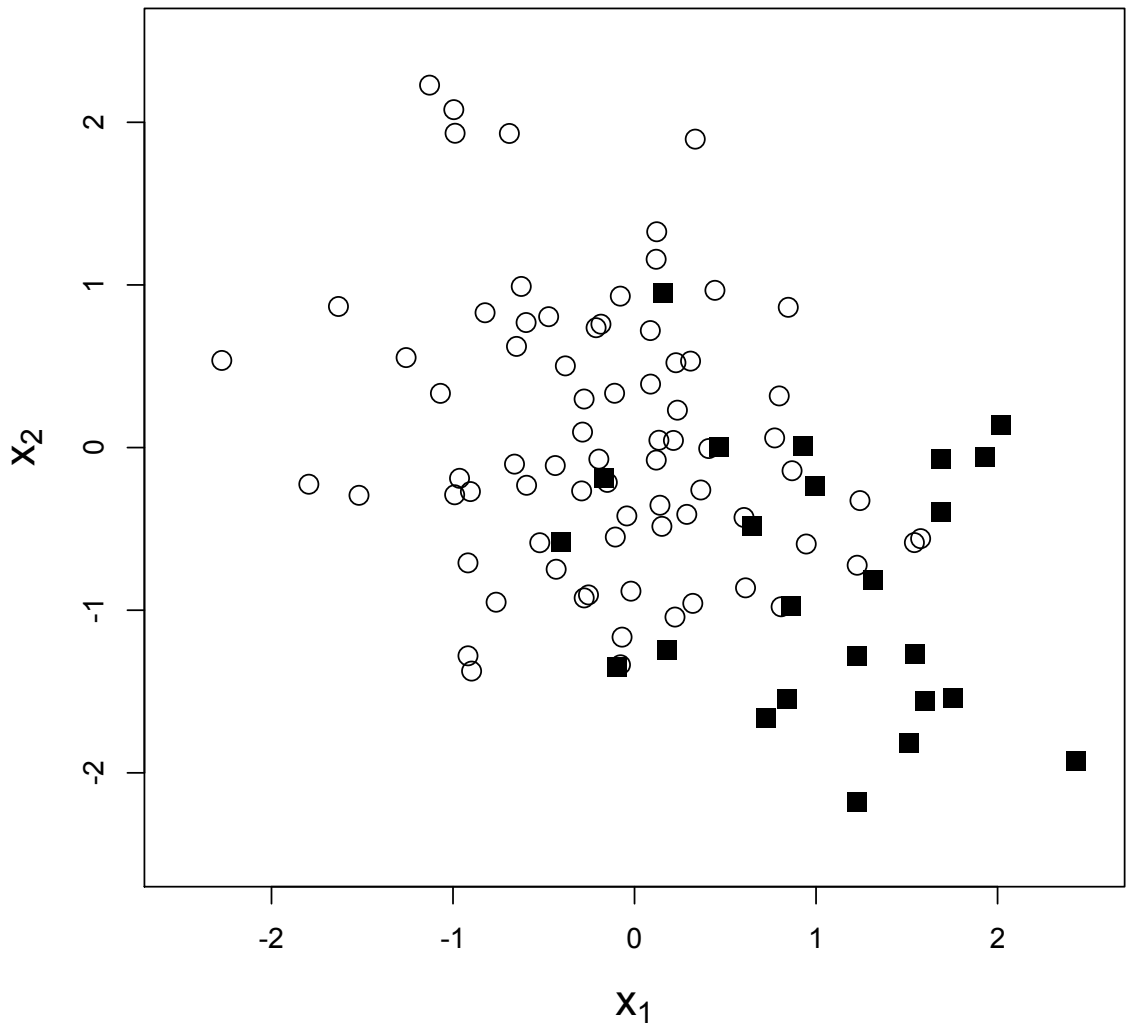
$$p \leftarrow p - (\varepsilon/2M)\frac{\partial U_0}{\partial q}$$

end for

$$p \leftarrow p - (\varepsilon/2)\frac{\partial U_1}{\partial q}$$

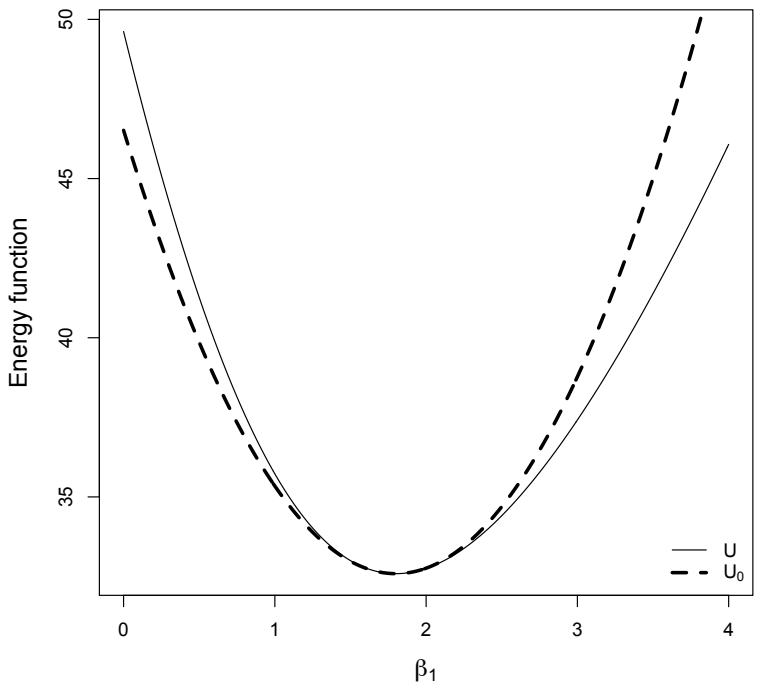
end for

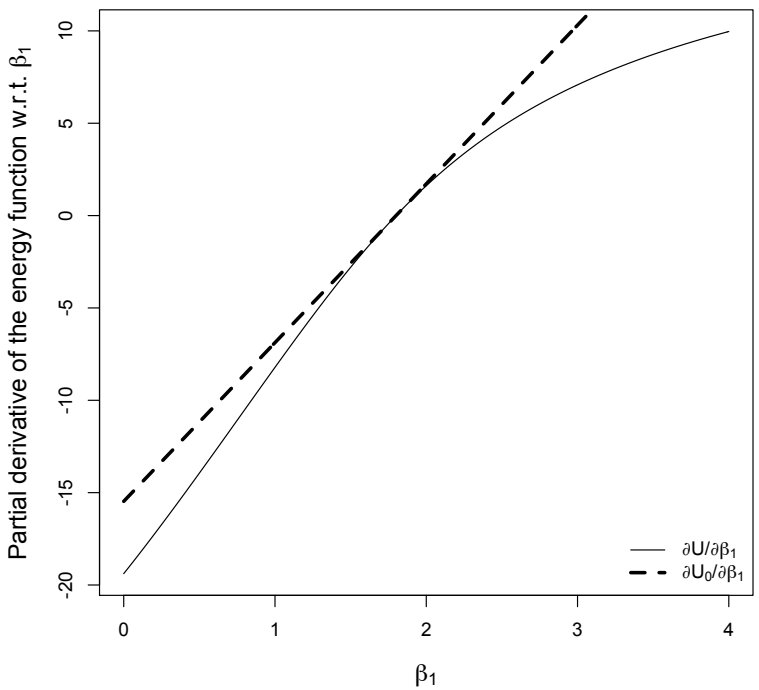
Split HMC for logistic regression



Split HMC for logistic regression

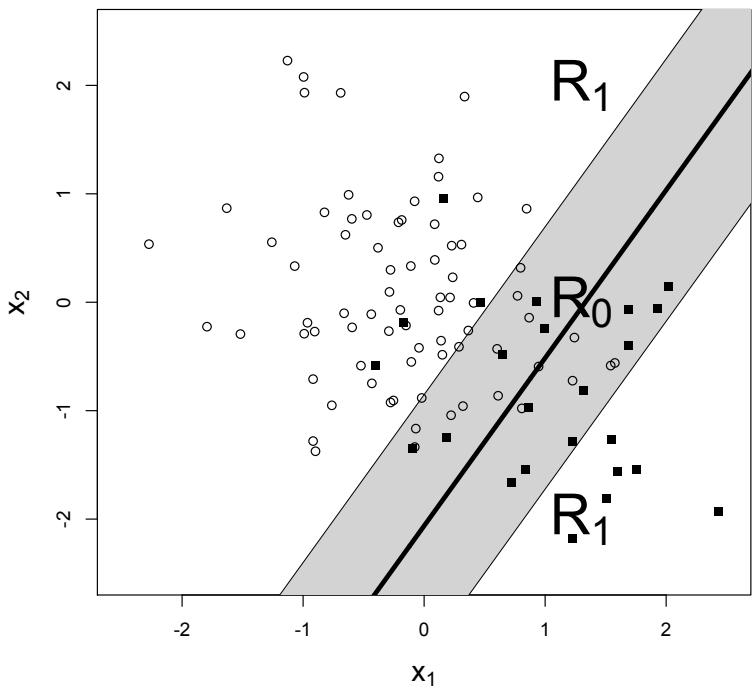
- For Algorithm 1, we use the MAP estimates, $\hat{\theta}$, for model parameters θ.
- $U_0(\theta)$ is the potential energy function of the normal distribution $N(\hat{\theta}, \mathcal{J}^{-1}(\hat{\theta}))$

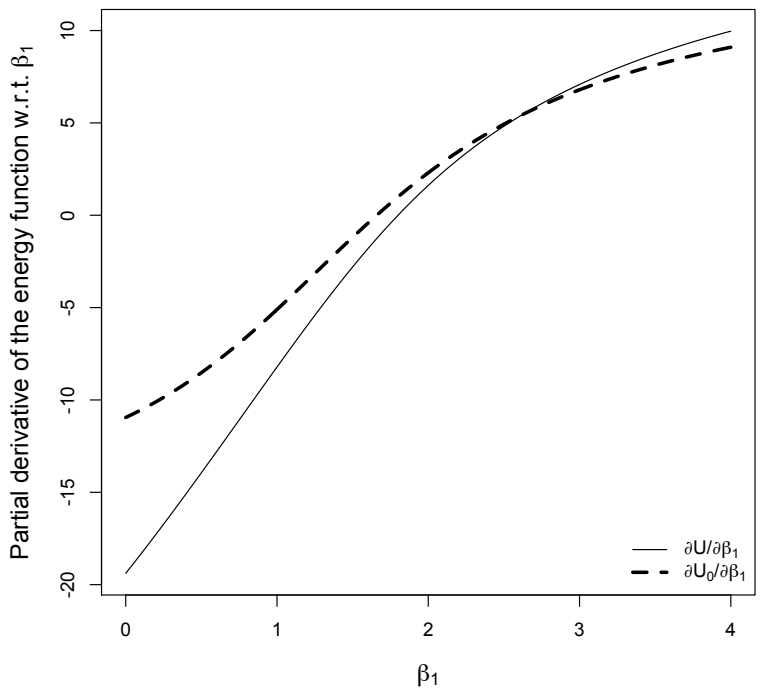




Split HMC for logistic regression

- For Algorithm 2, we use the MAP estimates, $\hat{\theta}$ and define U_0 based on the data points with high entropy.
- These are the points that are close to the classification boundary defined based on $\hat{\theta}$.





Split HMC for logistic regression

- Note that U_0 is not used to approximate U.
- Rather, $\partial U_0/\partial \beta_j$ is used to approximate $\partial U/\partial \beta_j$.
- Recall that

$$\frac{\partial U}{\partial \beta_j} = \frac{\beta_j}{\sigma_\beta^2} - \sum_{i=1}^{n} x_{ij}\left[y_i - \frac{\exp(\alpha + x_i^T\beta)}{1+\exp(\alpha + x_i^T\beta)}\right]$$

- The term $\exp(\alpha + x_i^T\beta)/(1+\exp(\alpha + x_i^T\beta))$ is in fact $P(y_i = 1|x_i, \alpha, \beta)$.
- For high entropy data points, this estimated probability is close to 0.5.

Experiments

- We set the number of leapfrog steps to $L = 20$ for the standard HMC, and find ε such that the acceptance probability (AP) is close to 0.65.
- We set L and ε for the Split HMC methods such that the trajectory length, εL, remains the same, but with a larger stepsize and hence a smaller number of steps.
- To measure the efficiency of each sampling method, we use the autocorrelation time (ACT).
- ACT can be roughly interpreted as the number of MCMC transitions required to produce samples that can be considered as independent.

Experiments– Simulated data

	HMC	Split HMC	
		Normal Appr.	Data Splitting
L	20	10	3
g	20	10	12.6
s	0.187	0.087	0.096
AP	0.69	0.74	0.74
τ	4.6	3.2	3.0
$\tau \times g$	92	**32**	38
$\tau \times s$	0.864	**0.284**	0.287

Experiments– StatLog

	HMC	Split HMC	
		Normal Appr.	Data Splitting
L	20	14	3
g	20	14	13.8
s	0.033	0.026	0.023
AP	0.69	0.74	0.85
τ	5.6	6.0	4.0
$\tau \times g$	112	84	**55**
$\tau \times s$	0.190	0.144	**0.095**

Experiments– Chess

	HMC	Split HMC	
		Normal Appr.	Data Splitting
L	20	9	2
g	20	13	11.8
s	0.022	0.011	0.013
AP	0.62	0.73	0.62
τ	10.7	12.8	12.1
$\tau \times g$	214	**115**	143
$\tau \times s$	0.234	**0.144**	0.161

Future directions

- Finding tractable approximations to the posterior distribution other than normal.
- Other methods for splitting the Hamiltonian dynamics by splitting the data,
- Combining our method with Riemannian Manifold HMC (Girolami and Calderhead, 2011) and No-U-Turn sampler (Hoffman and Gelman, 2011).