STATS271/371: Applied Bayesian Statistics

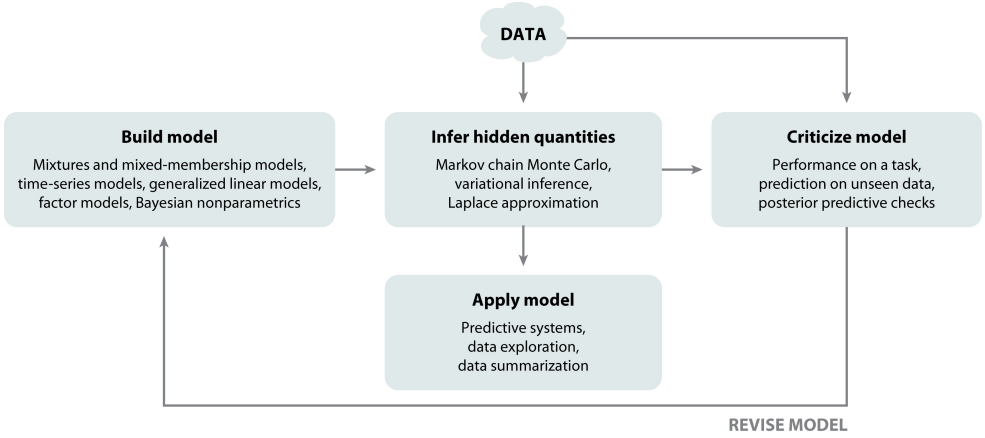
Robust regression models and Intro to MCMC

Scott Linderman

(Some slides are adapted from unpublished notes by David Blei.)

April 21, 2021

Box's Loop



Blei DM. 2014.
Annu. Rev. Stat. Appl. 1:203–32

Blei, *Ann. Rev. Stat. App.* 2014.

Lap 3: Robust regress models and Intro to MCMC

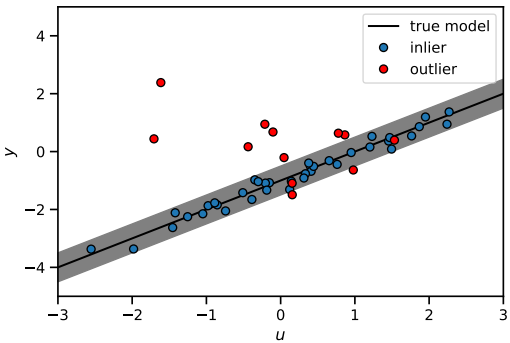
- **Model:** Robust regression models
- **Algorithm:** Markov chain Monte Carlo (specifically, HMC)
- **Criticism:** MCMC Diagnostics

Robustness to outliers

Consider the following model:

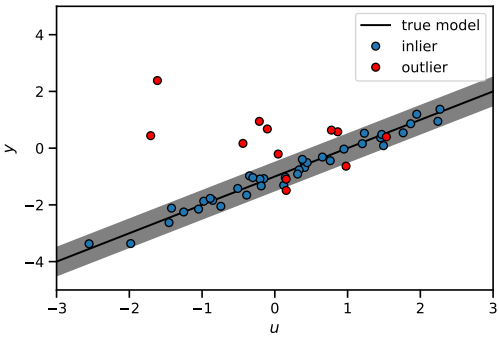
$$y_n \sim \begin{cases} \mathcal{N}(\boldsymbol{w}^\top \boldsymbol{x}_n, \sigma^2) & \text{with prob 0.85} \\ \mathcal{N}(0, \sigma^2_{\text{out}}) & \text{o.w.} \end{cases}$$

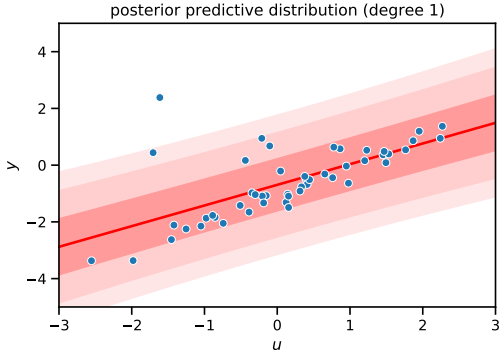
where $\sigma^2_{\text{out}} \gg \sigma^2$.



Robustness to outliers II

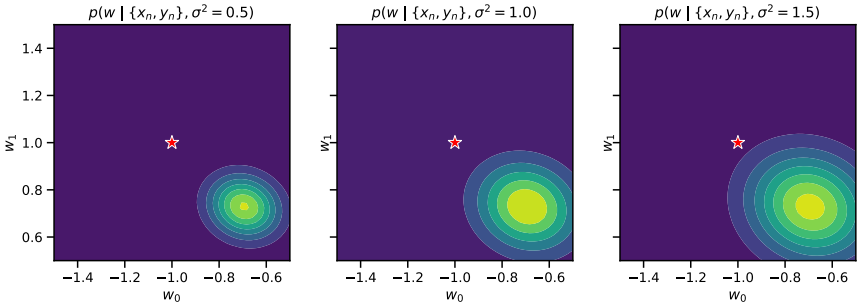
A standard Bayesian linear regression (Lap 1) compensates by inferring different weights and a larger variance.





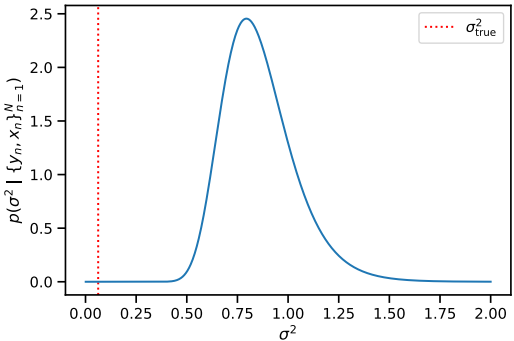
Robustness to outliers III

We also see this in the posterior distribution of the weights...



Robustness to outliers IV

...And in the posterior distribution of the variance



Robust modeling with heavy tailed distributions

One way to allow for outliers is via a discrete mixture distribution, like we used to simulate this data. More on this next week!

Mixtures are one way of creating *heavy tailed* conditional distributions. I.e. they place more probability mass on observations far from the mean or mode.

Heavier tails allow the model to effectively downweight some observations when inferring the weights. In the discrete mixture example, if we knew which data points came from the outlier distribution, we would simply ignore them (set their weights to zero) when calculating the posterior distribution of the regression coefficients.

Of course, we don't *a priori* which datapoints are inliers and which are outliers.

Robust inference largely amounts to *figuring out how to weight different observations* during inference.

Student's t as a scale-mixture of Gaussians

The Student's t distribution takes the discrete mixture idea to a continuous limit.

We can view the Student's distribution as a continuous *scale-mixture of Gaussians*,

$$t(y \mid \mu, \tau^2, \nu) = \int \mathcal{N}(y \mid \mu, \sigma^2)\, \text{Inv}{-}\chi^2(\sigma^2 \mid \nu, \tau^2)\, \mathrm{d}\sigma^2 \tag{1}$$

$$= \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu\tau^2}} \left(1 + \frac{1}{\nu\tau^2}(y-\mu)^2\right)^{-\frac{\nu+1}{2}} \tag{2}$$

Equivalently, we can think of $y \sim t(\mu, \tau^2, \nu)$ as arising from a two-step sampling procedure,

$$\sigma^2 \sim \text{Inv}{-}\chi^2(\nu, \tau^2) \tag{3}$$

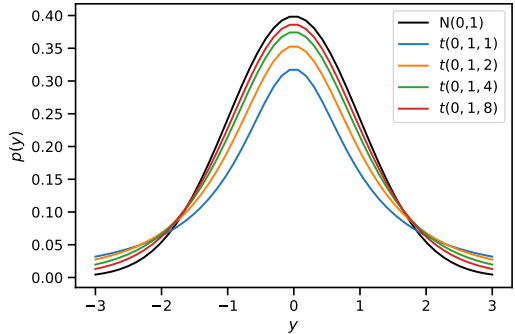
$$y \sim \mathcal{N}(\mu, \sigma^2). \tag{4}$$

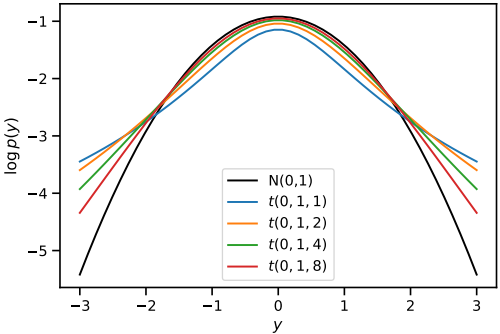
First sample a random variance, then sample a Gaussian observation.

Look familiar? Recall the posterior predictive distribution in Bayesian linear regression.

The Student's t distribution has heavy tails

As the degree of freedom parameter ν goes to infinity, the Student's t converges to a Gaussian.





Robust regression with Student's t noise model

Simple idea: replace the normal distribution in Bayesian linear regression with a Student's t.

$$y_n \sim t(\boldsymbol{w}^\top \boldsymbol{x}_n, \tau^2, \nu) \iff \sigma_n^2 \sim \text{Inv}{-}\chi^2(\nu, \tau^2) \tag{5}$$

$$y_n \sim \mathcal{N}(\boldsymbol{w}^\top \boldsymbol{x}_n, \sigma_n^2). \tag{6}$$

We can think of this as giving each datapoint its own random variance σ_n^2.

How does this relate to "reweighting" various datapoints? Suppose we somehow knew the variances σ_n^2 and just needed to infer the parameters $\boldsymbol{w}$. Recall that sufficient statistics for $\boldsymbol{w}$ where $\sum_{n=1}^N \frac{y_n \boldsymbol{x}_n}{\sigma_n^2}$, $\sum_{n=1}^N \frac{\boldsymbol{x}_n \boldsymbol{x}_n^\top}{\sigma_n^2}$, and $\sum_{n=1}^N \frac{1}{\sigma_n^2}$. Effectively, datapoints with larger variance contribute less to the sufficient statistics.

Probabilistic model

Unfortunately, we don't know the variances (knowing the variances is like knowing which datapoints are outliers) so we have to work with the t distribution instead.

Our likelihood is,

$$p(\{y_n\}_{n=1}^N \mid \boldsymbol{w}, \tau^2, \nu, \{\boldsymbol{x}_n\}_{n=1}^N) = \prod_{n=1}^N t(y_n \mid \boldsymbol{w}^\top \boldsymbol{x}_n, \tau^2, \nu) \tag{7}$$

$$= \prod_{n=1}^N \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu\tau^2}} \left(1 + \frac{1}{\nu\tau^2}\left(y_n - \boldsymbol{w}^\top \boldsymbol{x}_n\right)^2\right)^{-\frac{\nu+1}{2}} \tag{8}$$

This doesn't simplify, but it is differentiable and easy to compute any point $(\boldsymbol{w}, \tau^2, \nu)$.

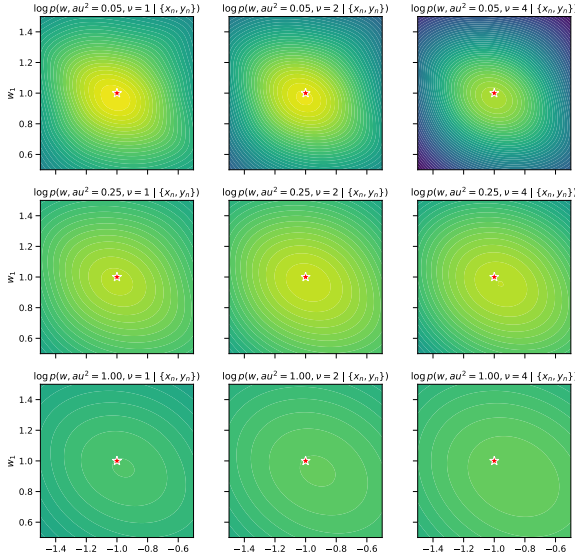
To complete the model, assume a broad Gaussian prior on the *log* of non-negative parameters,

$$p(\log \tau^2, \log \nu) = \mathcal{N}(\log \tau^2 \mid 0, 3)\, \mathcal{N}(\log \nu \mid 0, 3), \tag{9}$$

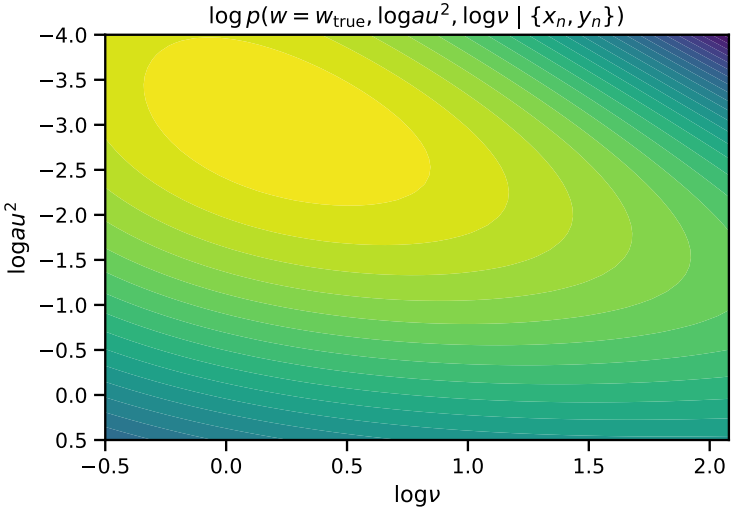
and an improper uniform prior on the weights.

A Gaussian prior on the log of a parameter is equivalent to a *log normal* prior on the parameter.

Visualizing the log joint probability

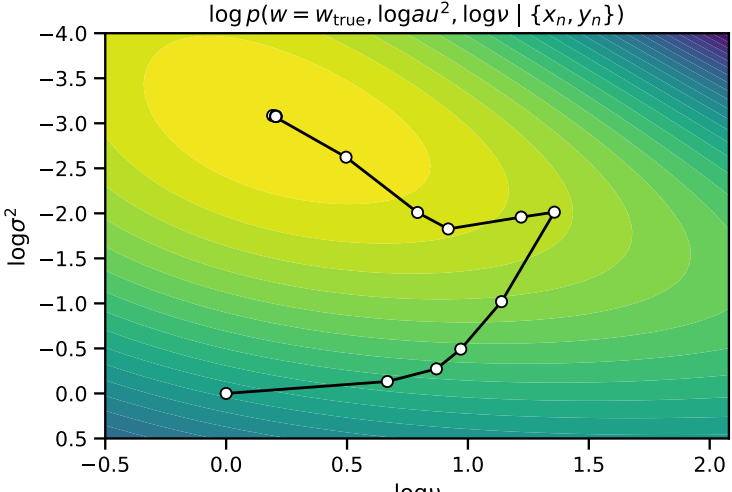


Visualizing the log joint probability II



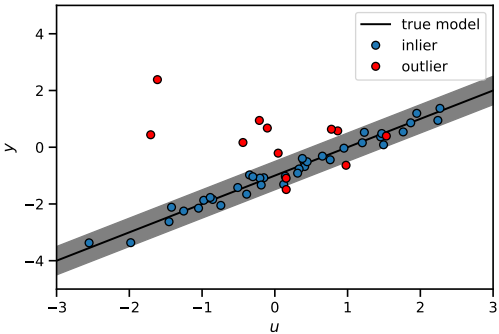
MAP Estimation

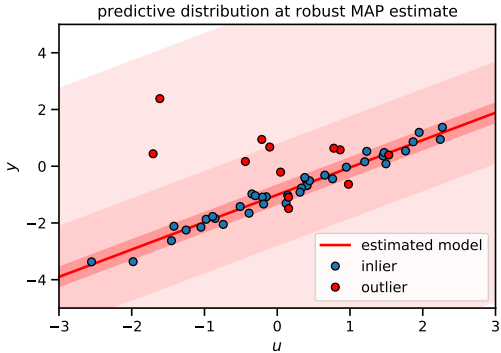
Again, the posterior doesn't have a simple analytical form, but we can still try our tools from last week.
E.g. we can use black box optimizers to find the mode.



Predictive distribution with w_{MAP}

The MAP estimate of the parameters yields a much better estimate of the weights by allowing for a heavy tailed distribution of outliers.





Notation

- Let $\boldsymbol{\theta} \in \Theta$ denote the model parameters.
  - In robust regression, $\boldsymbol{\theta} = (\boldsymbol{w}, \tau^2, \nu) \in \mathbb{R}^p \times \mathbb{R}_+ \times \mathbb{R}_+$.
- Let $\mathscr{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$ denote the observed data.

Posterior expectations

The central object of Bayesian inference is the posterior distribution, $p(\boldsymbol{\theta} \mid \mathscr{D})$.

However, we almost always interact with the posterior distribution through *expectations*.

- $\mathbb{E}_{p(\boldsymbol{\theta} \mid \mathscr{D})}[\boldsymbol{\theta}]$, the posterior mean.
- $\mathbb{E}_{p(\boldsymbol{\theta} \mid \mathscr{D})}[\mathbb{I}[\boldsymbol{\theta} \in \mathscr{A}]]$, the probability of the parameters being in set $\mathscr{A}$.
- $\mathbb{E}_{p(\boldsymbol{\theta} \mid \mathscr{D})}[p(\mathscr{D}' \mid \boldsymbol{\theta})]$, the posterior predictive density of new data $\mathscr{D}'$.

All of these can be written as $\mathbb{E}_{p(\boldsymbol{\theta} \mid \mathscr{D})}[f(\boldsymbol{\theta})]$ for some function f.

(One exception is the posterior mode, which is not so easily expressed as an expectation.)

Approximating posterior expectations

Generally, we can't analytically compute posterior expectations. (*Why not?*)

In these cases, we need to resort to approximations.

For example, we could use *quadrature methods* like Simpon's rule or the trapezoid rule to numerically approximate the integral over Θ.

Roughly,

$$\mathbb{E}_{p(\boldsymbol{\theta} \mid \mathscr{D})}[f(\boldsymbol{\theta})] \approx \sum_{m=}^{M} p(\boldsymbol{\theta}_m \mid \mathscr{D}) f(\boldsymbol{\theta}_m) \Delta_m \tag{10}$$

where $\boldsymbol{\theta}_m \subset \Theta$ is a grid of points and Δ_m is a volume around that point.

This works for low dimensional problems (say, up to 5 dimensions), but the number of points (M) needed to get a good estimate grows exponentially with the parameter dimension.

Monte Carlo approximations

Idea: approximate the expectation via sampling,

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathscr{D})}[f(\boldsymbol{\theta})] \approx \frac{1}{S}\sum_{s=1}^{S} f(\boldsymbol{\theta}_s) \quad \text{where} \quad \boldsymbol{\theta}_s \sim p(\boldsymbol{\theta} \mid \mathscr{D}). \tag{11}$$

Let $\hat{f} = \frac{1}{S}\sum_{s=1}^{S} f(\boldsymbol{\theta}_s)$ denote the Monte Carlo estimate. It is a random variable, since it's a function of random samples $\boldsymbol{\theta}_s$.

As such we can reason about its mean and variance. Clearly,

$$\mathbb{E}[\hat{f}] = \frac{1}{S}\sum_{s=1}^{S} \mathbb{E}_{p(\boldsymbol{\theta}|\mathscr{D})}[f(\boldsymbol{\theta})] = \mathbb{E}_{p(\boldsymbol{\theta}|\mathscr{D})}[f(\boldsymbol{\theta})]. \tag{12}$$

Thus, $\hat{f}$ is an *unbiased* estimate of the desired expectation.

Monte Carlo approximations II

What about its variance?

$$\mathrm{Var}[\hat{f}] = \mathrm{Var}\left(\frac{1}{S}\sum_{s=1}^{S} f(\boldsymbol{\theta}_s)\right) \tag{13}$$

$$= \frac{1}{S^2}\left(\sum_{s=1}^{S}\mathrm{Var}[f(\boldsymbol{\theta})] + 2\sum_{1\le s<s'\le S}\mathrm{Cov}[f(\boldsymbol{\theta}_s), f(\boldsymbol{\theta}_{s'})]\right) \tag{14}$$

If the samples are not only identically distributed but also *uncorrelated*, then $\mathrm{Var}[\hat{f}] = \frac{1}{S}\mathrm{Var}[f(\boldsymbol{\theta})]$.

In this case, the *root mean squared error* (RMSE) of the estimate is $\sqrt{\mathrm{Var}[\hat{f}]} = O(S^{-\frac{1}{2}})$.

Compare this to Simpson's rule, which for smooth 1D problems has error rate $O(S^{-4})$. That's roughly 8 times better!

However, for multidimensional problems, Simpson's rule is $O(S^{-\frac{4}{P}})$, whereas the **error rate of Monte Carlo does not depend on the dimensionality!**

The Catch

So far so good: we'll just draw a lot of samples to drive down our Monte Carlo error.

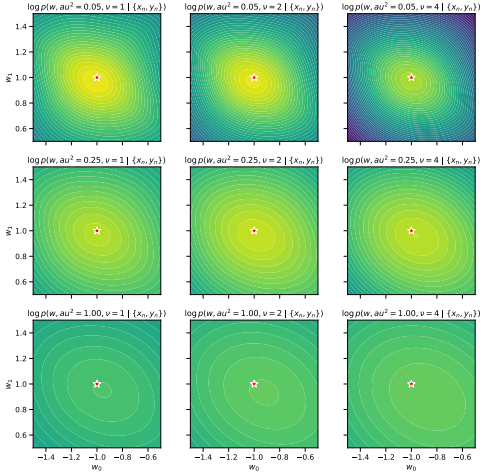
Here's the catch! How do you draw samples from the posterior $p(\boldsymbol{\theta} \mid \mathscr{D})$?

We're interested in Monte Carlo for cases where the posterior does not admit a simple closed form!

In general, sampling the posterior is as hard as computing the marginal likelihood.

Markov chain Monte Carlo (MCMC)

Idea: Design a Markov chain whose stationary distribution is the posterior.



Markov chains

A *Markov chain* is a joint distribution of a sequence of variables, $\pi(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_S)$.

(To avoid confusion with the model p, we denote the densities associated with the Markov chain by π.)

The Markov chain factorizes so that each variable is drawn conditional on the previous variable,

$$\pi(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_S) = \pi_1(\boldsymbol{\theta}_1) \prod_{s=2}^{S} \pi(\boldsymbol{\theta}_s \mid \boldsymbol{\theta}_{s-1}). \tag{15}$$

This is called the *Markov property*.

The distribution $\pi_1(\boldsymbol{\theta}_1)$ is called the *initial distribution*.

The distribution $\pi(\boldsymbol{\theta}_s \mid \boldsymbol{\theta}_{s-1})$ is called the *transition distribution*. If the transition distribution is the same for each s, the Markov chain is *homogenous*.

Stationary distributions

Let $\pi_s(\boldsymbol{\theta}_s)$ denote the marginal distribution of sample $\boldsymbol{\theta}_s$. It can be obtained recursively as,

$$\pi_s(\boldsymbol{\theta}_s) = \int \pi_{s-1}(\boldsymbol{\theta}_{s-1})\, \pi(\boldsymbol{\theta}_s \mid \boldsymbol{\theta}_{s-1})\, \mathrm{d}\boldsymbol{\theta}_{s-1}. \tag{16}$$

We are interested in the asymptotic behavior of the marginal distributions as $s \to \infty$.

A distribution $\pi^*(\boldsymbol{\theta})$ is a **stationary distribution** if,

$$\pi^*(\boldsymbol{\theta}) = \int \pi^*(\boldsymbol{\theta}')\, \pi(\boldsymbol{\theta} \mid \boldsymbol{\theta}')\, \mathrm{d}\boldsymbol{\theta}'. \tag{17}$$

That is, suppose the marginal of sample $\boldsymbol{\theta}'$ is $\pi^*(\boldsymbol{\theta})$. Then the marginal of the next time point is also $\pi^*(\boldsymbol{\theta})$.

Detailed balance

How can we relate transition distributions and stationary distributions?

A sufficient (but not necessary) condition for $\pi^*(\boldsymbol{\theta})$ to be a stationary distribution is that it satisfies *detailed balance*,

$$\pi^*(\boldsymbol{\theta}')\pi(\boldsymbol{\theta} \mid \boldsymbol{\theta}') = \pi^*(\boldsymbol{\theta})\pi(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) \tag{18}$$

In words, the probability of starting at $\boldsymbol{\theta}'$ and moving to $\boldsymbol{\theta}$ is the same as that of starting at $\boldsymbol{\theta}$ and moving to $\boldsymbol{\theta}'$, if you draw the starting point from the stationary distribution.

To see that detailed balance is sufficient, integrate both sides to get,

$$\int \pi^*(\boldsymbol{\theta}')\pi(\boldsymbol{\theta} \mid \boldsymbol{\theta}') \,\mathrm{d}\boldsymbol{\theta}' = \int \pi^*(\boldsymbol{\theta})\pi(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) \,\mathrm{d}\boldsymbol{\theta}' = \pi^*(\boldsymbol{\theta}) \int \pi(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) \,\mathrm{d}\boldsymbol{\theta}' = \pi^*(\boldsymbol{\theta}). \tag{19}$$

Thus, $\pi^*(\boldsymbol{\theta})$ is a stationary distribution of the Markov chain with transitions $\pi(\theta \mid \theta')$.

Ergodicity

Detailed balance can be used to show that $\pi^*(\boldsymbol{\theta})$ is *a* stationary distribution, but not that it is *the unique* one.

This is where *ergodicity* comes in. A Markov chain is ergodic if $\pi_s(\boldsymbol{\theta}_s) \rightarrow \pi^\star(\boldsymbol{\theta})$ regardless of $\pi_1(\boldsymbol{\theta}_1)$.

An ergodic chain has only one stationary distribution, $\pi^*(\boldsymbol{\theta})$.

The easiest way to prove ergodicity is to show that it is possible to reach any $\boldsymbol{\theta}'$ from any other $\boldsymbol{\theta}$. E.g. this is trivially so if $\pi(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) > 0$.

Note: a more technical definition is that all pairs of sets *communicate*, in which case the chain is *irreducible*, and that each state is *aperiodic*. The definitions can be a bit overwhelming.

The Metropolis-Hastings algorithm

Finally we come to our **main objective**: designing a Markov chain for which *the posterior is the unique stationary distribution*.

That is, we want $\pi^*(\boldsymbol{\theta}) = p(\boldsymbol{\theta} \mid \mathscr{D})$.

Recall our **constraint**: we can only compute the joint probability (the numerator in Bayes' rule), not the marginal likelihood (the denominator).

Fortunately, that still allows us to compute ratios of posterior densities! We have,

$$\frac{p(\boldsymbol{\theta} \mid \mathscr{D})}{p(\boldsymbol{\theta}' \mid \mathscr{D})} = \frac{p(\boldsymbol{\theta}, \mathscr{D})}{p(\mathscr{D})} \frac{p(\mathscr{D})}{p(\boldsymbol{\theta}', \mathscr{D})} = \frac{p(\boldsymbol{\theta}, \mathscr{D})}{p(\boldsymbol{\theta}', \mathscr{D})}. \tag{20}$$

The Metropolis-Hastings algorithm II

Now rearrange the detailed balance condition to relate ratios of transition probabilities to ratios of joint probabilities,

$$\frac{\pi(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}' \mid \boldsymbol{\theta})} = \frac{\pi^*(\boldsymbol{\theta})}{\pi^*(\boldsymbol{\theta}')} = \frac{p(\boldsymbol{\theta} \mid \mathscr{D})}{p(\boldsymbol{\theta}' \mid \mathscr{D})} = \frac{p(\boldsymbol{\theta}, \mathscr{D})}{p(\boldsymbol{\theta}', \mathscr{D})} \tag{21}$$

To construct such a transition distribution $\pi(\boldsymbol{\theta} \mid \boldsymbol{\theta}')$, break it down into two steps.

1. Sample a proposal $\boldsymbol{\theta}$ from a *proposal distribution* $q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')$,
2. Accept the proposal with *acceptance probability* $a(\boldsymbol{\theta}' \to \boldsymbol{\theta})$. (Otherwise, set $\boldsymbol{\theta} = \boldsymbol{\theta}'$.)

Thus,

$$\pi(\boldsymbol{\theta} \mid \boldsymbol{\theta}') = \begin{cases} q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')\, a(\boldsymbol{\theta}' \to \boldsymbol{\theta}) & \text{if } \boldsymbol{\theta}' \neq \boldsymbol{\theta} \\ \int q(\boldsymbol{\theta}'' \mid \boldsymbol{\theta}')\, (1 - a(\boldsymbol{\theta}' \to \boldsymbol{\theta}''))\, \mathrm{d}\boldsymbol{\theta}'' & \text{if } \boldsymbol{\theta}' = \boldsymbol{\theta} \end{cases} \tag{22}$$

The Metropolis-Hastings algorithm III

The constraint in (21) is trivially satisfied when $\boldsymbol{\theta} = \boldsymbol{\theta}'$. When $\boldsymbol{\theta} \neq \boldsymbol{\theta}'$, we need

$$\frac{\pi(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}' \mid \boldsymbol{\theta})} = \frac{q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')\, a(\boldsymbol{\theta}' \to \boldsymbol{\theta})}{q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})\, a(\boldsymbol{\theta} \to \boldsymbol{\theta}')} = \frac{p(\boldsymbol{\theta}, \mathscr{D})}{p(\boldsymbol{\theta}', \mathscr{D})} \Rightarrow \frac{a(\boldsymbol{\theta}' \to \boldsymbol{\theta})}{a(\boldsymbol{\theta} \to \boldsymbol{\theta}')} = \underbrace{\frac{p(\boldsymbol{\theta}, \mathscr{D})\, q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})}{p(\boldsymbol{\theta}', \mathscr{D})\, q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}}_{\triangleq A(\boldsymbol{\theta}' \to \boldsymbol{\theta})} \tag{23}$$

WLOG, assume $A(\boldsymbol{\theta}' \to \boldsymbol{\theta}) \leq 1$. (If it's not, its inverse $A(\boldsymbol{\theta} \to \boldsymbol{\theta}')$ must be.)

A simple way to ensure detailed balance is to set $a(\boldsymbol{\theta}' \to \boldsymbol{\theta}) = A(\boldsymbol{\theta}' \to \boldsymbol{\theta})$ and $a(\boldsymbol{\theta} \to \boldsymbol{\theta}') = 1$.

We can succinctly capture both cases with,

$$a(\boldsymbol{\theta}' \to \boldsymbol{\theta}) = \min\left\{1, A(\boldsymbol{\theta}' \to \boldsymbol{\theta})\right\} = \min\left\{1, \frac{p(\boldsymbol{\theta}, \mathscr{D})\, q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})}{p(\boldsymbol{\theta}', \mathscr{D})\, q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}\right\}. \tag{24}$$

The Metropolis algorithm

Now consider the special case in which the proposal distribution is symmetric; i.e. $q(\boldsymbol{\theta} \mid \boldsymbol{\theta}') = q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})$.

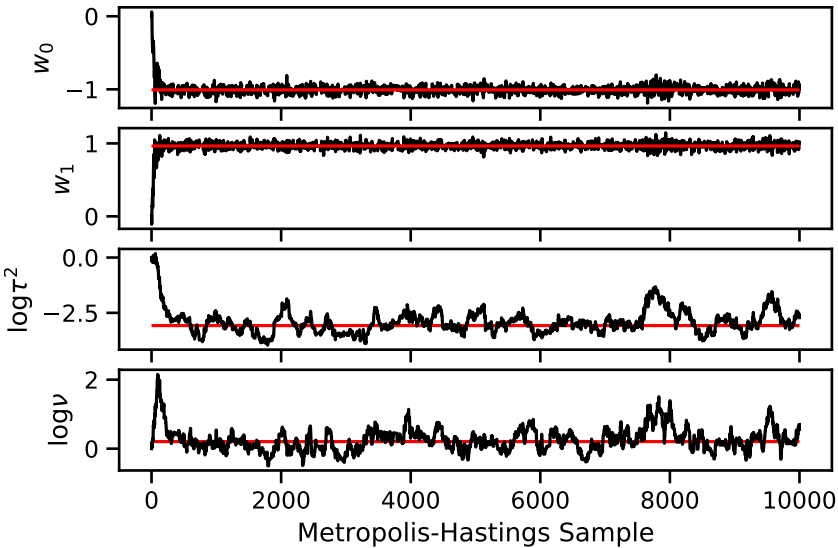
Then the proposal densities cancel in the acceptance probability and,

$$a(\boldsymbol{\theta}' \to \boldsymbol{\theta}) = \min \left\{ 1, \frac{p(\boldsymbol{\theta}, \mathscr{D})}{p(\boldsymbol{\theta}', \mathscr{D})} \right\}. \tag{25}$$

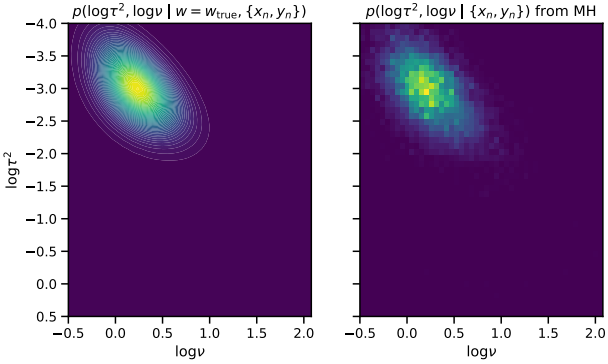
In other words, you accept any proposal that moves "uphill," and only accept "downhill" moves with some probability.

This is called the *Metropolis algorithm* and it has close connections to *simulated annealing*.

Metropolis-Hastings Trace Plot



Metropolis-Hastings Density Approximation



Autocovariance and autocorrelation

Recall that,

$$\mathrm{Var}[\hat{f}] = \frac{1}{S^2}\left(\sum_{s=1}^{S}\mathrm{Var}[f(\boldsymbol{\theta})] + 2\sum_{1\le s<s'\le S}\mathrm{Cov}[f(\boldsymbol{\theta}_s), f(\boldsymbol{\theta}_{s'})]\right) \tag{26}$$

$$\approx \frac{1}{S}\left(\mathrm{Var}[f(\boldsymbol{\theta})] + 2\sum_{\ell=1}^{S}\mathrm{Cov}[f(\boldsymbol{\theta}_s), f(\boldsymbol{\theta}_{s+\ell})]\right) \tag{27}$$

since the covariance is only a function of the lag ℓ once the chain has reached stationarity.

Note: At stationarity, the samples are identically distributed but still correlated!

$\mathrm{Cov}[f(\boldsymbol{\theta}_s), f(\boldsymbol{\theta}_{s+\ell})]$ is called the *autocovariance*. It's a function of the lag ℓ (and the function f).

The *autocorrelation function* (ACF) is defined as, $\mathrm{acf}_f(\ell) = \mathrm{Cov}[f(\boldsymbol{\theta}_s), f(\boldsymbol{\theta}_{s+\ell})]/\mathrm{Var}[f(\theta)]$ so

$$\mathrm{Var}[\hat{f}] \approx S^{-1}\,\mathrm{Var}[f(\boldsymbol{\theta})]\left(1 + 2\sum_{\ell=1}^{S}\mathrm{acf}_f[\ell]\right). \tag{28}$$

Effective sample size

The *effective sample size* (ESS) approximates the effective number of independent samples you get from an autocorrelated chain, in terms of the variance of the Monte Carlo estimate,

$$S_{\text{eff},f} = S\frac{\text{Var}[f(\theta)]}{\text{Var}[f(\theta)](1 + 2\sum_{\ell=1}^{\infty}\text{acf}_f[\ell])} = \frac{S}{1 + 2\sum_{\ell=1}^{\infty}\text{acf}_f[\ell]} \tag{29}$$

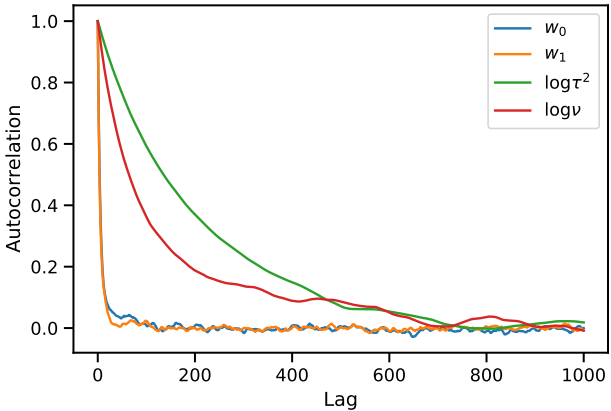
and we let S_{eff} denote the ESS of the identity functional $f(\theta) = \theta$.

You have to be a bit careful when estimating the ESS—for large values of ℓ the sample correlation is too noisy. Typically, we stop when the sample acf is negative. See Section 11.5 of BDA3 for more details.

In practice, there are already good implementations in Python (c.f. `tfp.stats.effective_sample_size`) and R (c.f. the `coda` package).

Autocorrelation of the MH trace above

We used `tfp.stats.auto_correlation` to compute the autocorrelation function for the traces above,



Effective sample sizes (per 100,000 samples): 5629, 7235, 257, and 364.

Autocorrelation of a Random Walk

[From Geyer, 2011] To get some intuition, consider the following Markov chain,

$$\theta_s = \rho\,\theta_{s-1} + \epsilon_s \tag{30}$$

where $\epsilon_s \sim \mathcal{N}(0, \tau^2)$.

MH isn't quite a mean-reverting random walk, but that's not a terrible model.

In this toy example, we can calculate the autocovariance of the identity functional $f(\theta) = \theta$,

$$\mathrm{Cov}[\theta_s, \theta_{s+\ell}] = \rho\mathrm{Cov}[\theta_s, \theta_{s+\ell-1}] = \rho^{\ell-1}\mathrm{Cov}[\theta_s, \theta_{s+1}] = \rho^{\ell}\mathrm{Var}[\theta_s] \tag{31}$$

so the autocorrelation function decays geometrically as $\mathrm{acf}(\ell) = \rho^{\ell}$.

At stationarity,

$$\mathrm{Var}[\theta_s] = \mathrm{Var}[\theta_{s+1}] = \rho^2\mathrm{Var}[\theta_s] + \tau^2 \Rightarrow \mathrm{Var}[\theta_s] = \frac{\tau^2}{1-\rho^2} \tag{32}$$

For $\rho^2 < 1$, the stationary distribution exists and is $\mathcal{N}(0, \frac{\tau^2}{1-\rho^2})$.

Autocorrelation of a Random Walk II

Letting $f(\theta) = \theta$, we have,

$$\mathrm{Var}[\hat{f}] = \frac{1}{S}\left(\mathrm{Var}[\theta] + 2\sum_{\ell=1}^{S}\mathrm{Cov}[\theta_s, \theta_{s+\ell}]\right) \tag{33}$$

$$= \frac{1}{S}\cdot\mathrm{Var}[\theta]\left(1 + 2\sum_{\ell=1}^{S}\rho^{\ell}\right) \tag{34}$$

$$\approx \frac{1}{S}\cdot\mathrm{Var}[\theta]\left(1 + 2\frac{\rho}{1-\rho}\right) \qquad \text{(for large } S\text{)} \tag{35}$$

$$= \frac{1}{S}\cdot\mathrm{Var}[\theta]\cdot\frac{1+\rho}{1-\rho} \tag{36}$$

and

$$S_{\text{eff}} = S\cdot\frac{1-\rho}{1+\rho}. \tag{37}$$

As $\rho \to 0$, we recover ordinary Monte Carlo. As $\rho \to 1$, the autocorrelation causes the variance of the estimator to blow up and the effective sample size to go to zero.

What about bias?

The mean squared error (MSE) of the estimator is determined by both the bias and the variance.

Ordinary Monte Carlo estimates are unbiased by construction, but MCMC estimates are only *asymptotically unbiased*.

Bias is introduced whenever the initial distribution $\pi_1(\boldsymbol{\theta})$ differs from the stationary distribution; i.e. in all practical cases!

Fortunately, the bias decays as $O(S^{-1})$ whereas the variance decays as $O(S^{-1/2})$, so asymptotically the MSE is dominated by the variance.

How can we make smarter proposals?

Metropolis-Hastings with a symmetric Gaussian proposal behaves like a random walk.

Neal [2012] argues that in D dimensions, random walk MH needs $O(D^2)$ iterations to get an independent sample.

Can we develop more efficient transition distributions?

Yes, if we have more information about the log probability.

For example, suppose that the log probability $\log p(\theta)$ is differentiable. We can use the gradient to make proposals that move farther and are more likely to be accepted.

Metropolis Adjusted Langevin Algorithm (MALA)

The *Metropolis-Adjusted Langevin Algorithm* uses the gradient of the log probability to make asymmetric proposals,

$$q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} + \tau \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}, \mathcal{D}),\, 2\tau^2 \boldsymbol{I}) \tag{38}$$

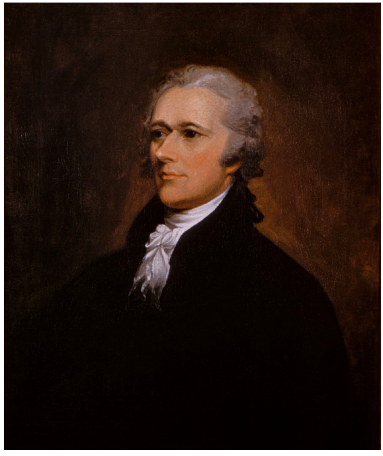
Note: $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) \neq q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')$! To calculate the acceptance probability, you need the gradient at both points.

MALA can be motivated as a discrete-time approximation to the *Langevin* diffusion, a continuous-time stochastic differential equation for modeling molecular dynamics.

In high dimensions, the extra information provided by the gradient can lead to much more efficient chains. Neal argues that MALA needs $O(D^{4/3})$ computation to produce an independent sample.

But why stop at one gradient step?

Three Hamiltons



Hamiltonian Monte Carlo

Reference: Neal [2012] *MCMC using Hamiltonian dynamics.*

Idea: *Think of negative log probability as an energy landscape. Now imagine a puck sliding around on this bumpy surface. Give it random kicks; it will tend to slide downhill toward points of low potential energy (high probability). Each kick can displace the puck by a large amountDone properly, the puck will visit points with probability proportional to the posterior probability.*

Notation

Following Neal [2012], let

- $\boldsymbol{q} \in \mathbb{R}^D$ denote the *position*; i.e. the current parameters (previously θ)
- $\boldsymbol{p} \in \mathbb{R}^D$ denote the *momentum*; auxiliary variables that we don't care about, but which are necessary for HMC.
- $\boldsymbol{z} = [\boldsymbol{q}, \boldsymbol{p}]^\top \in \mathbb{R}^{2D}$ denote the combined *state of the system*.
- $\boldsymbol{M}$ denote the *mass matrix*, another artificial construct. Typically, this will be $m\boldsymbol{I}$
- $U(\boldsymbol{q})$ denote the *potential energy*
- $K(\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^\top \boldsymbol{M}^{-1}\boldsymbol{p}$ denote the *kinetic energy*

Hamiltonian dynamics

The *Hamiltonian* is the sum of the potential $H(\boldsymbol{q}, \boldsymbol{p}) = U(\boldsymbol{q}) + K(\boldsymbol{p})$.

The partial derivatives determine how the state evolves over time,

$$\frac{\mathrm{d}q_d}{\mathrm{d}t} = \frac{\partial H}{\partial p_d} = [\boldsymbol{M}^{-1}\boldsymbol{p}]_d \tag{39}$$

$$\frac{\mathrm{d}p_d}{\mathrm{d}t} = -\frac{\partial H}{\partial q_d} = -\frac{\partial U}{\partial q_d} \tag{40}$$

for $d = 1, \ldots, D$.

Compactly,

$$\frac{\mathrm{d}\boldsymbol{z}}{\mathrm{d}t} = J\nabla H(z) \tag{41}$$

where

$$J = \begin{bmatrix} \mathbf{0}, \boldsymbol{I} \\ -\boldsymbol{I}, \mathbf{0} \end{bmatrix} \tag{42}$$

One dimensional example

Consider the case where $D = 1$ and $U(q) = \frac{1}{2}q^2$ and $K(p) = \frac{1}{2}p^2$.

The partial derivatives are

$$\frac{\partial H}{\mathrm{d}p} = p \tag{43}$$

$$-\frac{\partial H}{\mathrm{d}q} = -q \tag{44}$$

so

$$\frac{\mathrm{d}\boldsymbol{z}}{\mathrm{d}t} = J\boldsymbol{z}. \tag{45}$$

This is a linear dynamical system, and the state at time $t + \Delta t$ is $\boldsymbol{z}(t + \Delta t) = e^{J\Delta t}\boldsymbol{z}(t)$.

Since $J\Delta t$ is skew-symmetric, the matrix exponential $e^{J\Delta t}$ is orthogonal. More precisely, $\boldsymbol{z}(t + \Delta t)$ is a rotation about the origin of $\boldsymbol{z}(t)$.

Properties of Hamiltonian dynamics

1. **Reversibility:** The mapping from $z(t) \to z(t + \Delta t)$ is one-to-one and invertible. To go from $z(t + \Delta t)$ to $z(t)$, negate $p(t + \Delta t)$, apply the the Hamiltonian dynamics for Δt time, and negate the momentum again.

2. **Conservation of energy:** The Hamiltonian (which is the total energy in a closed system) is conserved,

$$\frac{\mathrm{d}H}{\mathrm{d}t} = \sum_{d=1}^{D} \frac{\mathrm{d}q_d}{\mathrm{d}t} \frac{\partial H}{\partial q_d} + \frac{\mathrm{d}p_d}{\mathrm{d}t} \frac{\partial H}{\partial p_d} \tag{46}$$

$$= \sum_{d=1}^{D} \frac{\partial H}{\partial p_d} \frac{\partial H}{\partial q_d} - \frac{\partial H}{\mathrm{d}q_d} \frac{\partial H}{\partial p_d} = 0. \tag{47}$$

Properties of Hamiltonian dynamics II

3 **Volume preserving:** A set R in (q, p) space will have the same volume after being mapped through Hamiltonian dynamics. This follows from the fact that the divergence of the vector field is zero everywhere:

$$\operatorname{div} \frac{\mathrm{d}\boldsymbol{z}}{\mathrm{d}t} = \sum_{d=1}^{D} \frac{\partial}{\partial q_d} \frac{\mathrm{d}q_d}{\mathrm{d}t} + \frac{\partial}{\partial p_d} \frac{\mathrm{d}p_d}{\mathrm{d}t} = \sum_{d=1}^{D} \frac{\partial}{\partial q_d} \frac{\partial H}{\partial p_d} - \frac{\partial}{\partial p_d} \frac{\partial H}{\partial q_d} = \sum_{d=1}^{D} \frac{\partial^2 H}{\partial q_d \partial p_d} - \frac{\partial^2 H}{\partial q_d \partial p_d} = 0. \tag{48}$$

4 **Sympleticness** Let B be the Jacobian of the transformation from $\boldsymbol{z}(t) \rightarrow \boldsymbol{z}(t + \Delta t)$. It turns out that,

$$B^\top J^{-1} B = J^{-1} \tag{49}$$

which implies that $|B^\top||J^{-1}||B| = |J^{-1}|$ and thus $|B| = 1$. I.e. the dynamics preserve volume.

Discretizing Hamilton's equations

The properties above apply to the *continuous time* Hamiltonian dynamics. Can we maintain them in practice?

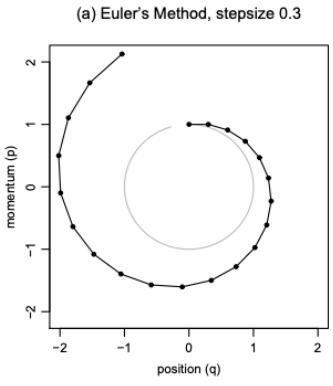
Idea: In practice, to simulate Δt elapsed time, we break it down into steps of size $\Delta t/\epsilon$.

Euler's method: Update the state as,

$$\boldsymbol{z}(t+\epsilon) = \boldsymbol{z}(t) + \epsilon \left.\frac{\mathrm{d}\boldsymbol{z}}{\mathrm{d}t}\right|_{\boldsymbol{z}(t)} \tag{50}$$

$$\Rightarrow p_d(t+\epsilon) = p_d(t) - \epsilon \left.\frac{\partial U}{\partial q_d}\right|_{\boldsymbol{q}(t)} \tag{51}$$

$$q_d(t+\epsilon) = q_d(t) + \epsilon \frac{p_d(t)}{m_d} \tag{52}$$



Simple Euler integration does not preserve volume: trajectories eventually diverge, even with small ϵ.

The Leapfrog Integrator

Instead, alternate updates of p and q

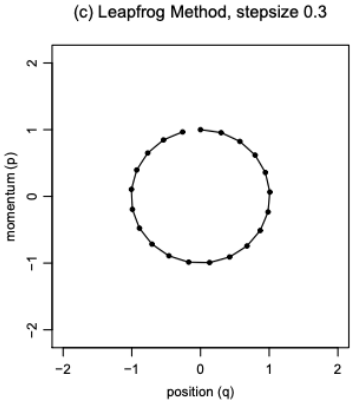
$$p_d(t+\tfrac{\epsilon}{2}) = p_d(t) - \frac{\epsilon}{2}\frac{\partial U}{\partial q_d}\Big|_{\boldsymbol{q}(t)} \tag{53}$$

$$q_d(t+\epsilon) = q_d(t) + \epsilon\frac{p_d(t)}{m_d} \tag{54}$$

$$p_d(t+\epsilon) = p_d(t+\tfrac{\epsilon}{2}) - \frac{\epsilon}{2}\frac{\partial U}{\partial q_d}\Big|_{\boldsymbol{q}(t+\epsilon)} \tag{55}$$

$$\tag{56}$$

Each update is a *shear transformation* in which only some variables change, by amounts that depend on the other, fixed variables. The determinant of such a transformation is one, so it preserves volume.



Using Hamiltonian dynamics for posterior inference

Define a joint distribution on positions and momenta as,

$$p(\boldsymbol{q}, \boldsymbol{p}) \propto \exp\left\{-H(\boldsymbol{q}, \boldsymbol{p})\right\} \propto \exp\left\{-U(\boldsymbol{q}) - K(\boldsymbol{p})\right\}. \tag{57}$$

Now let $U(\boldsymbol{q}) = -\log p(\boldsymbol{\theta} = \boldsymbol{q}, \mathscr{D})$ be the *negative* log joint probability. Then,

$$p(\boldsymbol{q}, \boldsymbol{p}) = p(\boldsymbol{\theta} = \boldsymbol{q} \mid \mathscr{D}) \times p(\boldsymbol{p}) \tag{58}$$

Samples of $\boldsymbol{q}$ will be marginally distributed according to the posterior $p(\boldsymbol{\theta} = \boldsymbol{q} \mid \mathscr{D})$.

Samples of $\boldsymbol{p}$ will be marginally distributed $p(\boldsymbol{p}) = \frac{\exp\{-K(\boldsymbol{p})\}}{\int_{\mathbb{R}^D} \exp\{-K(\boldsymbol{p})\}\, \mathrm{d}\boldsymbol{p}}$. These are *auxiliary variables* that we don't really care about—they're just there to help us construct MH proposals.

We choose $K(\boldsymbol{p})$ so $p(\boldsymbol{p})$ is convenient; e.g. if $K(\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^\top \boldsymbol{M}^{-1}\boldsymbol{p}$ then

$$p(\boldsymbol{p}) = \mathscr{N}(\boldsymbol{p} \mid \boldsymbol{0}, \boldsymbol{M}). \tag{59}$$

Hamiltonian Monte Carlo (HMC)

Hamiltonian Monte Carlo (HMC) is Metropolis-Hastings on the joint distribution of $(\boldsymbol{q}, \boldsymbol{p})$ with proposals based on Hamiltonian dynamics.

Starting at point $(\boldsymbol{q}', \boldsymbol{p}')$, sample the proposal distribution:

1. Throw away $\boldsymbol{p}'$ and sample new momenta from their marginal distribution $\boldsymbol{p} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{M})$.
2. Approximate Hamiltonian dynamics on $(\boldsymbol{q}, \boldsymbol{p})$ for Δt time using $L = \Delta t / \epsilon$ Leapfrog steps each of size ϵ. Call the resulting point $(\boldsymbol{q}, \boldsymbol{p})$.
3. Flip the momentum $\boldsymbol{p} \leftarrow -\boldsymbol{p}$ to make the proposal symmetric.

Then accept the proposed point $(\boldsymbol{q}, \boldsymbol{p})$ with probability,

$$a((\boldsymbol{q}', \boldsymbol{p}') \to (\boldsymbol{q}, \boldsymbol{p})) = \min \left\{ 1, \frac{\exp\{-H(\boldsymbol{q}, \boldsymbol{p})\}\, q(\boldsymbol{q}', \boldsymbol{p}' \mid \boldsymbol{q}, \boldsymbol{p})}{\exp\{-H(\boldsymbol{q}', \boldsymbol{p}')\} q(\boldsymbol{q}, \boldsymbol{p} \mid \boldsymbol{q}', \boldsymbol{p}')} \right\} = \min \left\{ 1, \frac{\exp\{-H(\boldsymbol{q}, \boldsymbol{p})\}}{\exp\{-H(\boldsymbol{q}', \boldsymbol{p}')\}} \right\}. \quad (60)$$

If the Hamiltonian dynamics were simulated exactly, HMC would always accept. In practice, differences arise from numerical integration errors.

HMC Dynamics on a Correlated 2D Gaussian

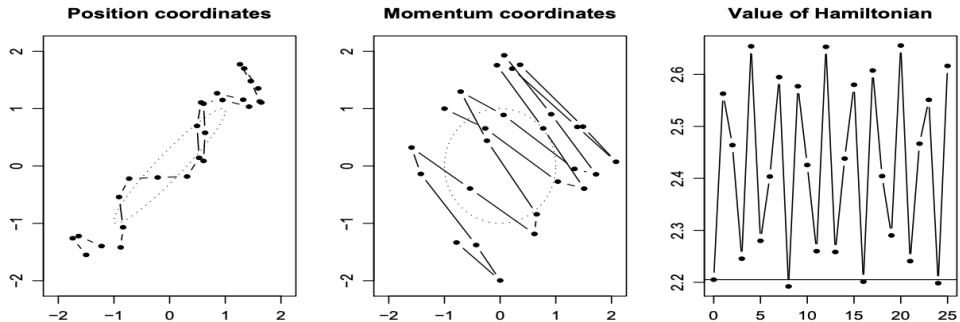


Figure 3: A trajectory for a 2D Gaussian distribution, simulated using 25 leapfrog steps with a stepsize of 0.25. The ellipses plotted are one standard deviation from the means. The initial state had $q = [-1.50, -1.55]^T$ and $p = [-1, 1]^T$.

HMC vs Random Walk MH

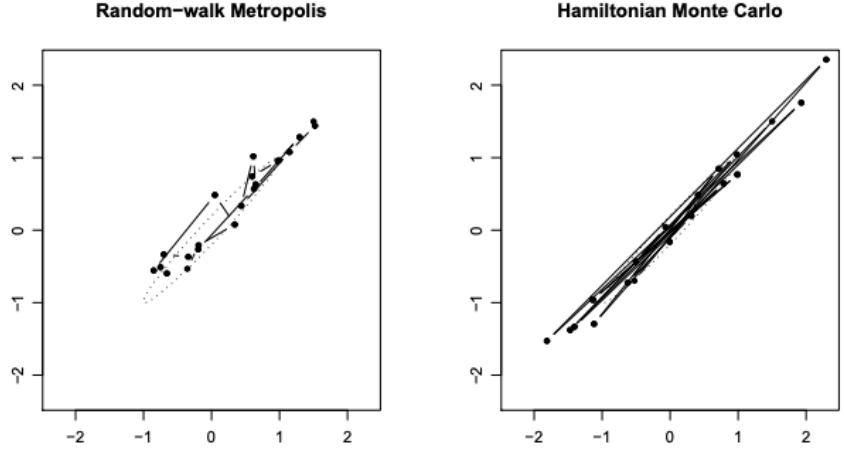


Figure 4: Twenty iterations of the random-walk Metropolis method (with 20 updates per iteration) and of the Hamiltonian Monte Carlo method (with 20 leapfrog steps per trajectory) for a 2D Gaussian distribution with marginal standard deviations of one and correlation 0.98. Only the two position coordinates are plotted, with ellipses drawn one standard deviation away from the mean.

HMC vs Random Walk MH II

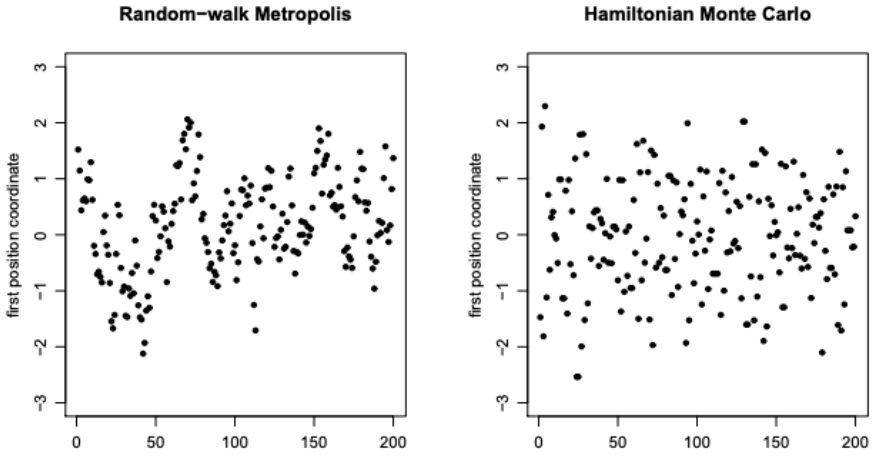


Figure 5: Two hundred iterations, starting with the twenty iterations shown above, with only the first position coordinate plotted.

HMC vs Random Walk MH in 100D

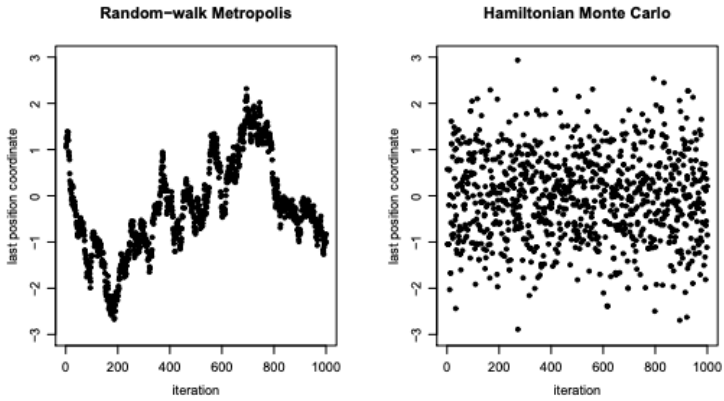
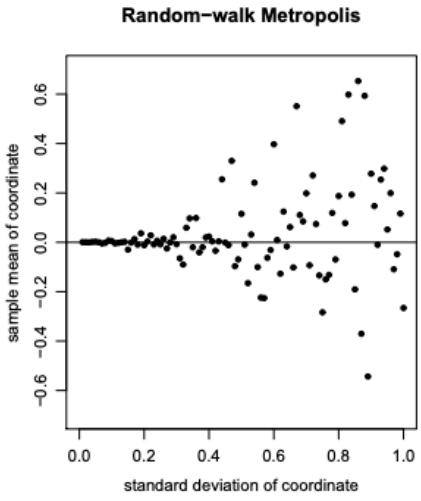
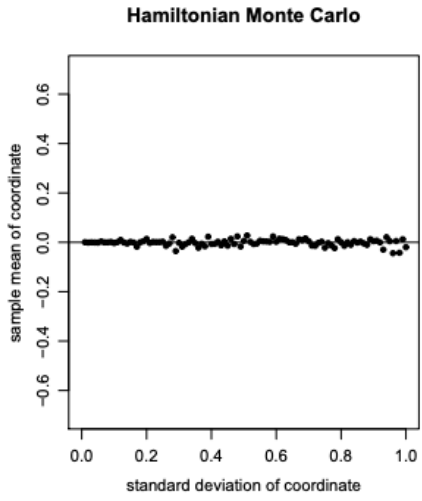


Figure 6: Values for the variable with largest standard deviation for the 100-dimensional example, from a random-walk Metropolis run and an HMC run with $L = 150$. To match computation time, 150 updates were counted as one iteration for random-walk Metropolis.

HMC vs Random Walk MH in 100D II





Benefits of avoiding random walks

- To maintain reasonably high acceptance probability, random walk MH needs proposal standard deviation (s.d.) comparable to the s.d. in the most constrained dimension (0.14 in the 2D Gaussian example and 0.01 in the 100D example).
- Num. iterations needed for RW-MH to reach an approximately independent state is proportional to the *square* of the largest standard deviation to the smallest; i.e. to the condition number of the covariance matrix.
- In contrast, integrating the Hamiltonian makes many steps in the same direction. The number of integration steps to reach an independent state is about the ratio of the largest s.d. to the smallest; i.e. the square root of the condition number.
- Neal [2012] argues that the number of leapfrog updates to reach an independent point scales as $O(D^{5/4})$, better than the $O(D^2)$ and $O(D^{4/3})$ estimates for random walk MH and MALA, respectively.
- However, we still need to tune the step size ϵ to be comparable to the smallest s.d.

Adapting the step size

- A simple strategy is to tune the step size adaptively during the initial run of the Markov chain.
- For example, set a target acceptance rate (Neal argues that it should be around 0.65), then increase the step size if you're accepting too often and decrease if you're rejecting too often.
- Andrieu and Thoms [2008] proposed a widely-used multiplicative update scheme; it is the default in `tfp.mcmc.SimpleStepSizeAdaptation`.
- The **No U-Turn Sampler (NUTS)** [Hoffman and Gelman, 2014] adapts the distance traveled in response to the curvature of the target density. Conceptually, it continues until the trajectory turns back on itself (hence the name, "No U-Turn")
- More details can be found in Betancourt [2017].

Demos

https://chi-feng.github.io/mcmc-demo/app.html

References I

Radford M Neal. MCMC using Hamiltonian dynamics. June 2012.

Christophe Andrieu and Johannes Thoms. A tutorial on adaptive MCMC. *Stat. Comput.*, 18(4):343–373, December 2008.

Matthew D Hoffman and Andrew Gelman. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

Michael Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. January 2017.