Have I Converged?

Diagnosis and Remediation

Bob Carpenter

Columbia University

Stan 2.17 (January 2018) http://mc-stan.org

Computation Target

Expectations of Function of R.V.

- Suppose $f(\theta)$ is a function of random variable vector θ
- Suppose the density of θ is $p(\theta)$
  - *Warning*: θ overloaded as random and bound variable
- Then $f(\theta)$ is also random variable, with expectation

$$\mathbb{E}[f(\theta)] = \int_{\Theta} f(\theta)\ p(\theta)\ d\theta.$$

  - where Θ is support of $p(\theta)$ (i.e., $\Theta = \{\theta \mid p(\theta) > 0\}$

QoI as Expectations

- Most Bayesian quantities of interest (QoI) are expectations over the posterior $p(\theta \mid y)$ of functions $f(\theta)$
- **Bayesian parameter estimation**: $\hat{\theta}$
  - $\hat{\theta} = \mathbb{E}[\theta|y]$ minimizes expected square error
- **Bayesian parameter (co)variance estimation**: $\text{var}[\theta \mid y]$
  - $\text{var}[\theta \mid y] = \mathbb{E}[\ (\theta - \hat{\theta})^2 \mid y]$
  - $\text{covar}[\theta_1, \theta_2 \mid y] = \mathbb{E}[(\theta_1 - \hat{\theta}_1)(\theta_2 - \hat{\theta}_2) \mid y]$
- **Bayesian event probability**: $\Pr[A \mid y]$
  - $\Pr[A \mid y] = \mathbb{E}[\ \text{I}[\theta \in A] \mid y]$
  - e.g., $\Pr[\theta_1 > \theta_2 \mid y] = \mathbb{E}[\ \text{I}[\theta_1 > \theta_2] \mid y]$

Expectations via Monte Carlo

- Generate draws $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(M)}$ drawn from $p(\theta)$
- Monte Carlo Estimator **plugs in average** for expectation:

$$\mathbb{E}[f(\theta)|y] \approx \frac{1}{M} \sum_{m=1}^{M} f(\theta^{(m)})$$

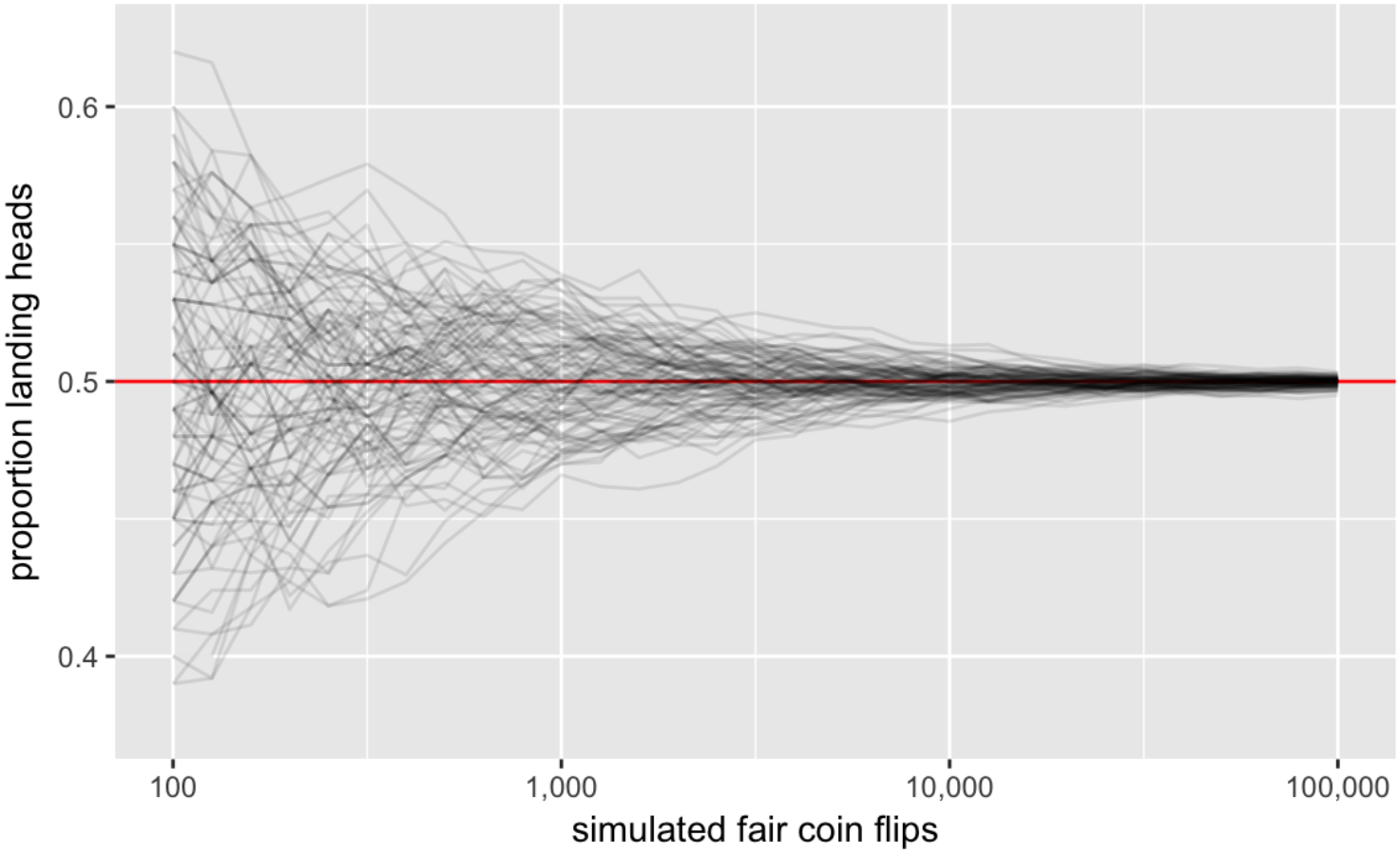
- Can be made **as accurate as desired**, because

$$\mathbb{E}[f(\theta)] = \lim_{M \to \infty} \frac{1}{M} \sum_{m=1}^{M} f(\theta^{(m)})$$

- *Reminder*: By CLT, error goes down as $1 / \sqrt{M}$

MCMC CLT

Central Limit Theorem (picture)



- proportion heads for 100 sequences of 100,000 flips
- converges gradually to expected value of 0.5

Central Limit Theorem (words)

- **The** theorem of statistics
  - Cardano (1501–1576) conjectured convergence; (Jacob) Bernoulli (1713) proved convergence for binomials (law of large numbers); de Moivre (1733) conjectured the CLT; Laplace (1812) proved i.i.d. version; Lyapunov (1901) removed i.i.d. constraint
- Sample **mean** of N i.i.d. variables with finite expectation
  - **converges** to their expectation as $N \to \infty$
  - **rate** of convergence is $\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$
  - constant factor determined by standard deviation
- Each decimal place of accuracy requires $100\times$ more draws

Central Limit Theorem (math)

- Simple i.i.d. version—can be established more generally
- Given N i.i.d. variables $\theta_1, \ldots, \theta_N$ with
  - $\mathbb{E}[\theta_n] = \mu$
  - $\text{sd}[\theta_n] = \sigma$

  the **central limit theorem** states

$$\lim_{N \to \infty} \frac{\theta_1 + \cdots + \theta_N}{N} \sim \text{Normal}\left(\mu, \frac{\sigma}{\sqrt{N}}\right)$$

Markov Chain Monte Carlo

- Simulating independent draws from the posterior $p(\theta|y)$ usually intractable
- Simulating a Markov chain $\theta^{(1)}, \ldots, \theta^{(M)}$ with marginals equal to posterior, i.e.,

$$p(\theta^{(m)}|y) = p(\theta|y)$$

  often is tractable

- Replace indepedent draws with Markov chain of draws
  - Plug in just like ordinary (non-Markov chain) Monte Carlo
  - Adjust standard errors for correlation in Markov chain

MCMC Central Limit Theorem

- Adjust standard errors for correlation in Markov chain

$$n\_eff = num\_iterations / adjustment\_for\_correlation$$

- With anti-correlated chains, n_eff can be larger than num_itera
- NUTS can produce anti-correlated chains

Geometric Ergodicity

- Geometric convergence in total variation of distribution
- Maximum event probability difference decreases geometricallyis target distribution

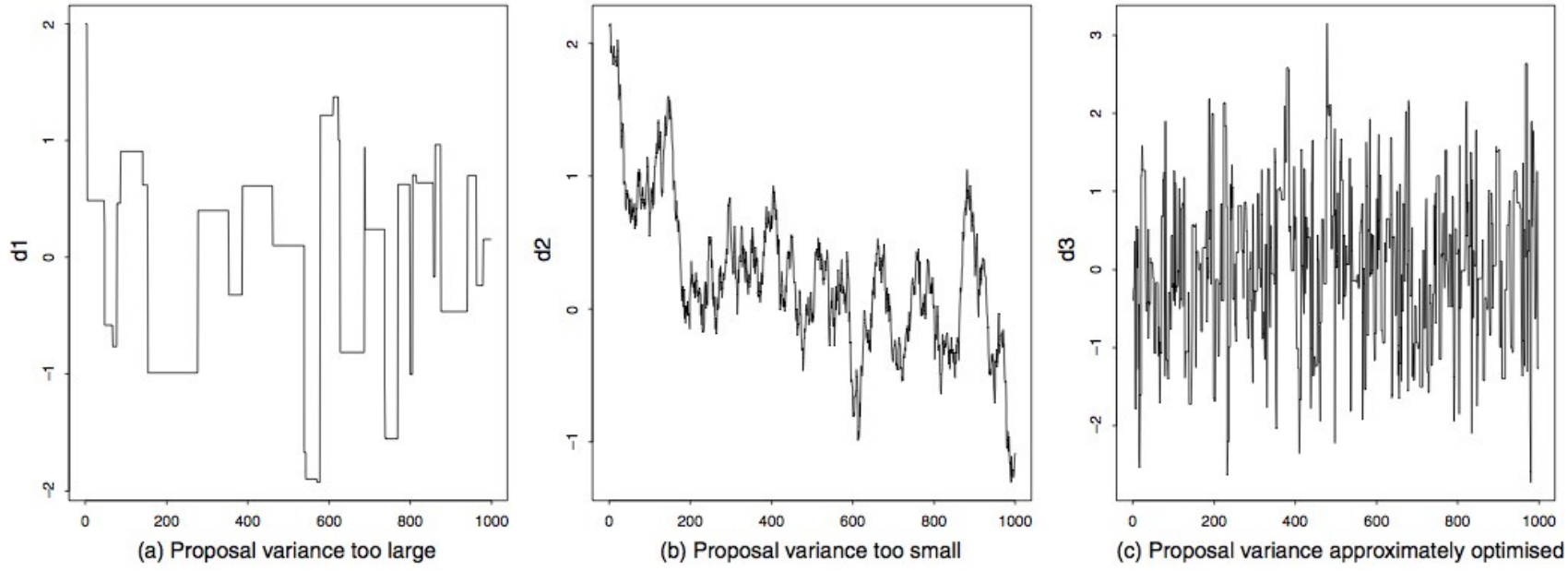
$$||P^n(\theta, \cdot) - \pi(\cdot)||_{var} \leq C_\theta \, \rho^n$$

  - where $\rho < 1$, C_θ is a constant, π is target distribution, and $P^n(\theta, \cdot)$ is the distribution starting at θ and taking n steps
  - Total variation normal

$$||\Pr||_{var} = \max_{A \subseteq \Omega} \Pr[A]$$

Optimal Proposal Scale?

- Proposal scale σ is a free; too low or high is inefficient

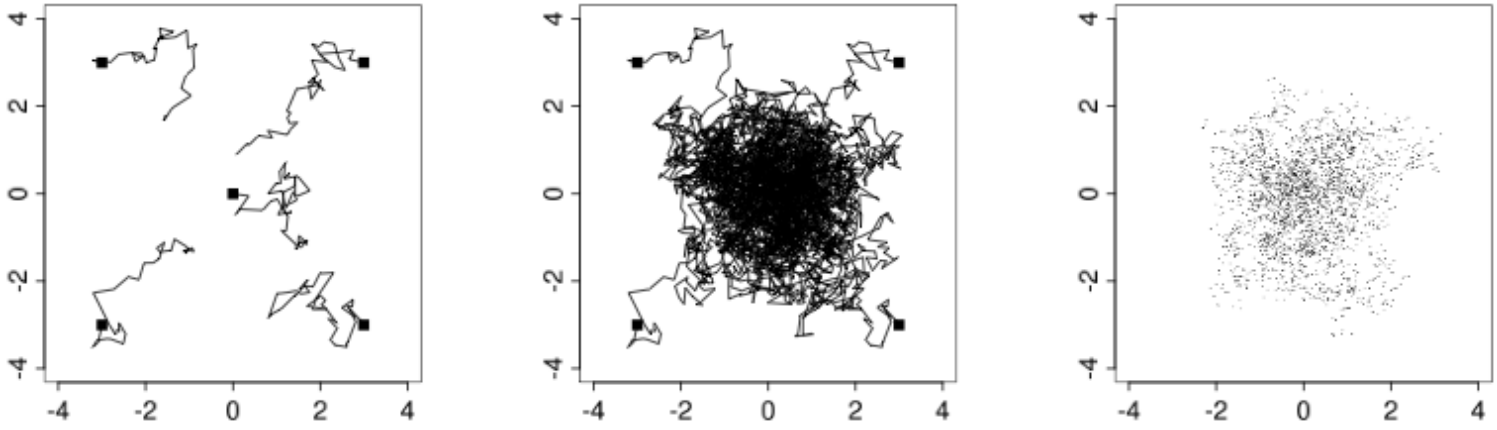


- *Traceplots* show parameter value on y axis, iterations on x
- Empirical tuning problem; theoretical optima exist for some cases

Roberts and Rosenthal (2001) Optimal Scaling for Various Metropolis-Hastings Algorithms. *Statistical Science*.

Convergence

- May take many iterations for chain to reach equilibrium
- Different initializations should converge in distribution



- Four chains with different starting points. *Left*) 50 iterations; *Center*) 1000 iterations; *Right*) Draws from second half of each chain

Stationarity

- Stationarity ensures $\theta^{(m)}$ and $\theta^{(m+n)}$ are identically distributed
  - usually not independent (vs. (anti-)correlated)
- Want convergence to posterior, so $\theta^{(m)}$ distributed as posterior $p(\theta|y)$

Potential Scale Reduction ($\hat{R}$)

- Gelman & Rubin recommend M chains of N draws with **diffuse initializations**
- Measure that each chain has same posterior mean and variance
- If not, may be stuck in multiple modes or just not converged yet
- Define statistic $\hat{R}$ of chains such that **at convergence**, $\hat{R} \rightarrow 1$
  - $\hat{R} >> 1$ implies non-convergence
  - $\hat{R} \approx 1$ **does not guarantee convergence**
  - Only measures marginals

Numerical Analysis

Floating-Point Standard: IEEE 754

- **Finite numbers** (s: sign; c: mantissa; q: exponent)

$$x = (-1)^s \times c \times 2^q$$

size	s, c *bits*	q *bits*	*range*	*precision*
32-bit	24	8	$\pm 3.4 \times 10^{38}$	7.2 digits
64-bit	53	11	$\pm 1.8 \times 10^{308}$	16 digits

- Quiet and signaling **not-a-number** (NaN)
- Positive and negative **infinity** ($+\infty, -\infty$)
- **Stan** uses 64-bit floating point

Catastrophic Cancellation

- Subtraction risks **catastrophic cancellation**
- Consider $0.99802 - 0.99801 = 0.00001$
  - input has five digits of precision
  - output has single digit of precision
- E.g., problem for sample variance of sequence x

$$\text{var}(x) = \frac{1}{N-1} \sum_{n=1}^{N} (x_n - \overline{x})^2$$

if elements x_n close to sample mean

$$\overline{x} = \frac{1}{N} \sum_{n=1}^{N} x_n$$

Welford's Algorithm

- **Streaming computation** uses fixed memory

```
N = 0;    mean = 0;    sum_sq_err = 0

handle(y):
    N += 1
    diff = y - mean
    mean = mean + diff / N
    diff2 = y - mean
    sum_sq_err += diff * diff2

mean():  return mean

var():  return sum_sq_err / (N - 1)
```

- Two stage difference is **less prone to cancellation**

Gaps Between Numbers

- Smallest number greater than zero
  - single precision: 1.4×10^{-45}
  - double precision: 4.9×10^{-324}
- Largest number less than one
  - single precision: $1 - 10^{-7.2}$
  - double precision: $1 - 10^{-16}$
- Gap size **depends on scale**

Lack of Transitivity

- For real numbers $x, y, z \in \mathbb{R}$,

$$x + (y + z) = (x + y) + z$$

- This can fail for floating point due to rounding
  - `(1 + 6e-17) + 6e-17 == 1`
  - `1 + (6e-17 + 6e-17) != 1`
- For square matrices $L L^\top$ is symmetric
- This won't hold for efficient matrix multiplications
  - `(L * L')[1, 2] != (L * L')[2, 1]`

Rounding and Equality

- Dangerous to compare floating point numbers
  - they may have lost precision during calculation
- Rounding
  - default: round toward nearest
  - round toward zero, round to plus or minus infinity

Overflow and Rounding

- Because there is a max size, operations can overflow
  - e.g., `exp(1000)`, `1e200 * 1e200`, ...
- Because there are gaps, operations can round to zero
  - e.g., `exp(-1000)`, `1e-200 * 1e-200`, ...
  - e.g., evaluating $\prod_{n=1}^{N} p(y_n|\theta)$ underflows for $N = 2000$ if $p(y_n|\theta) < 0.1$.

Example: `log1p` and CCDFs

- `log1p(x)` is for evaluating $\log$ near one
  - when x is near zero, `1 + x` catastrophically rounds to 1
  - this forces `log(1 + x)` to round to 0
  - `log1p(x)` avoids `1 + x` operation
  - `log1p(x)` uses Taylor series expansion of $\log(1 + x)$
- Complementary CDFs evaluate CDFs with values near one
  - X is some random variable, e.g., $X \sim \text{Normal}(0, 1)$
  - CDF: $F_X(x) = \Pr[X \leq x]$
  - CCDF: $F_X^{\complement}(x) = 1 - \Pr[X \leq x]$
  - converts range around one to range around zero

Example: log and log_sum_exp

- **Multiplication on the log scale**: log
  - $\log(a \times b) = \log a + \log b$
  - log converts multiplication to addition
  - $\log \prod_n x_n = \sum_n \log x_n$
  - avoids underflow and overflow even if $x_n \ll 1$ or $x_n \gg 1$
  - useful absolutely everywhere (e.g., log likelihoods)
- **Addition on the log scale**: log_sum_exp
  - $\log(a + b) = \log(\exp(\log a) + \exp(\log b))$
  - log converts addition to log sum of exponentials
  - avoids underflow and overflow, preserves precision
  - useful for mixtures (e.g., HMMs, zero-inflated Poisson)

Example: `log_sum_exp`

- Without loss of generality, assume $a > b$ (otherwise swap)

$$\begin{aligned}
\text{log_sum_exp}(a, b) &= \log(\exp(a) + \exp(b)) \\
&= a + \log(\exp(a - a) + \exp(b - a)) \\
&= a + \log(1 + \exp(b - a)) \\
&= a + \text{log1p}(\exp(\text{b} - \text{a}))
\end{aligned}$$

  - **increase precision**: pull a out of $\log()$ and $\exp()$
  - **increase precision**: use `log1p`
  - **prevents overflow**: can't overflow because $b - a \leq 0$

- Generalize to more than two inputs: subtract max

The Curse of Dimensionality

The Curse

- Intuitions formed in low dimensions **do not generalize**
- In high dimensions, **everything is far away**
  - random draws are far away from each other
  - random draws are far away from the mode or meaan
- Sampling algorithms that work in low dimensions often **fail in high dimensions**

Hyperballs in Hypercubes

- **sample uniformly** from container (square, cube, ...)
- 2 dimensions (x, y): compute Pr$[X^2 + Y^2 \leq 1]$
  - unit **disc** inscribed in square
  - calculate π given known area of circle (2π)
- 3 dimensions (x, y, z): compute Pr$[X^2 + Y^2 + Z^2 \leq 1]$
  - unit **ball** inscribed in cube
- N-dimensions $(x_1, \ldots, x_N)$: compute Pr$[X_1^2 + \cdots X_N^2 \leq 1]$
  - unit **hyperball** inscribed in hypercube
- Code event probability as **expectation of indicator**

Hyperballs in Hypercubes in Stan

```
generated quantities {
  int<lower=0, upper=1> in_ball[10];
  {
    real len_sq = 0;
    for (n in 1:10) {
      len_sq = len_sq + uniform_rng(-1, 1)^2;
      in_ball[n] = (len_sq <= 1);
    }
  }
}
```

- draw $x_1, \ldots, x_N$ is implicit in `uniform_rng`
- `in_ball[n]` is 1 iff $x_1^2 + \cdots + x_n^2 \leq 1;$ coded as indicator (`len <= 1`)
- sum of squares accumulation reduces quadratic time to linear

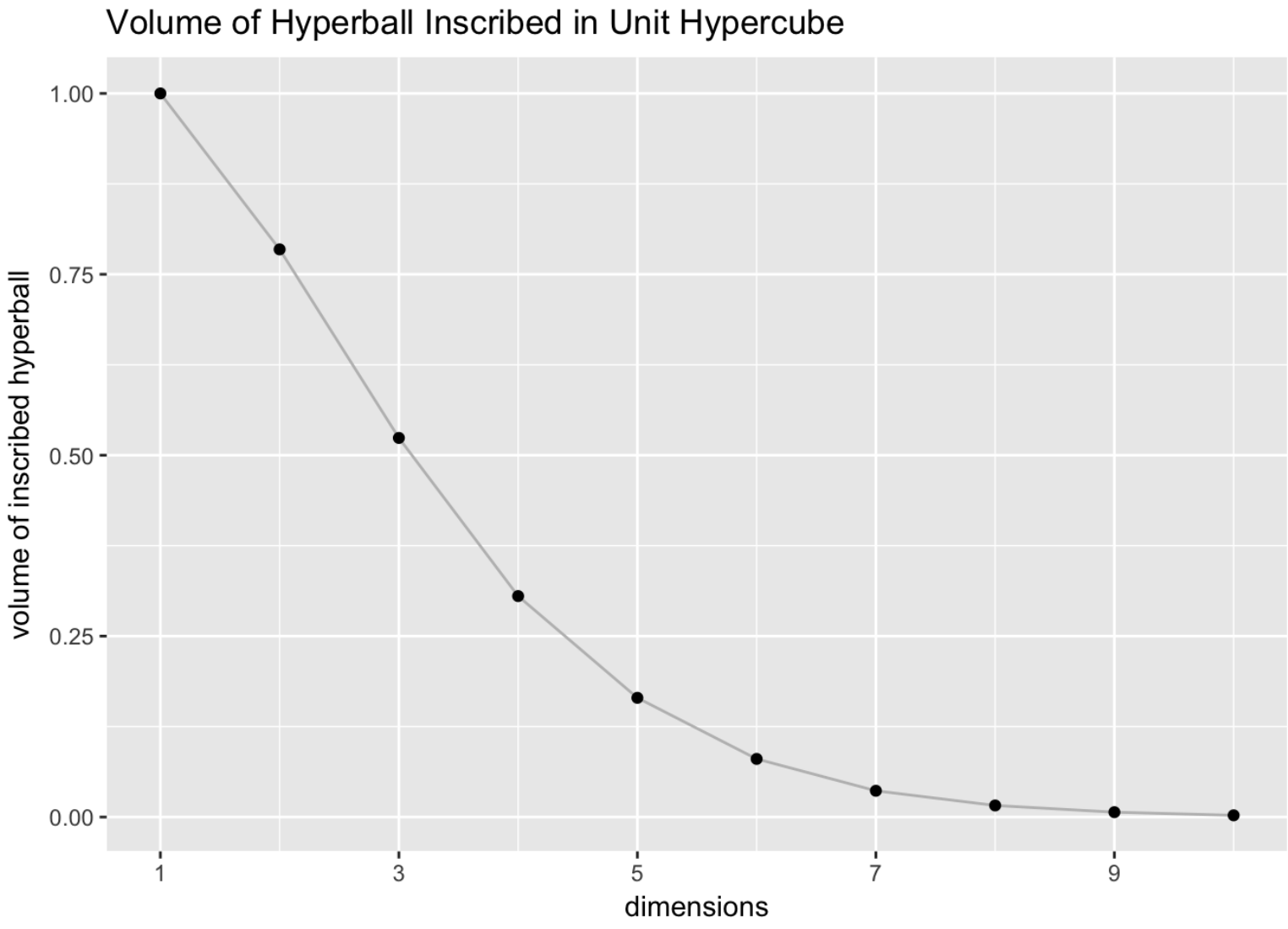
Hyperballs in Hypercubes in RStan

```
> fit <- stan("hyperballs.stan", algorithm="Fixed_param",
              iter=1e4)

> print(fit, probs=c())

             mean se_mean   sd n_eff Rhat
in_ball[1]   1.00       0 0.00 20000  NaN
in_ball[2]   0.78       0 0.41 20000    1
in_ball[3]   0.52       0 0.50 20000    1
in_ball[4]   0.31       0 0.46 20000    1
in_ball[5]   0.17       0 0.38 20000    1
in_ball[6]   0.08       0 0.27 20000    1
in_ball[7]   0.04       0 0.19 18460    1
in_ball[8]   0.02       0 0.12 19370    1
in_ball[9]   0.01       0 0.08 20000    1
in_ball[10] 0.00       0 0.05 20000    1
```

Proportion Volume in Hyperball



Typical Sets

Typical Set Example (1)

- Consider a game of chance with an 80% chance of winning
- Play the game 100 times independently

- What is most likely outcome?

Typical Set Example (2)

- For each trial, there is a 80% chance of success
- For each trial, most likely outcome is success
- Overall, the single **most likely outcome** is all successes

- What's the **most likely number of successes**?

Typical Set Example (3)

- Let $y_n \sim \text{Bernoulli}(0.9)$ for $n \in 1:100$ be the trials
- Expected number of successes

$$\begin{aligned}
\mathbb{E}\left[\sum_{n=1}^{100} y_n\right] &= \sum_{n=1}^{100} \mathbb{E}[y_n] \\
&= \sum_{n=1}^{100} 0.8 \\
&= 0.8 \times 100 \\
&= 80
\end{aligned}$$

- **most likely outcome** (all successes) **is an outlier**!

$$\Pr[100 \text{ successes}] = 0.8^{100} < 10^{-10}$$

Typical Set Example (4)

- Maximum likelihood (most likely) outcome is **atypical**
- Expectations involve count times probability
- 100 success sequences: $\binom{100}{100} = \frac{100!}{100! \times 1!} = 1$
- 80 success sequences: $\binom{100}{80} = \frac{100!}{80! \times 20!} > 10^{20}$
- Thus chance of 80 success is much higher than 100

$$\begin{aligned} \text{Binomial}(80 \mid 100, 0.8) &= \binom{100}{20} \times 0.8^{80} \times 0.2^{20} \\ &\gg \binom{100}{1} \times 0.8^{100} \\ &= \text{Binomial}(100 \mid 100, 0.8) \end{aligned}$$

Typical Set

- Goal is to **evaluate posterior expectations using draws**

$$\mathbb{E}\left[f(\theta) \mid y\right] = \int_{\Theta} f(\theta)\, p(\theta|y)\, \mathrm{d}\theta$$

$$\approx \frac{1}{M} \sum_{m=1}^{M} f(\theta^{(m)})$$

- A **typical set** A_ϵ (at some level) is the set
  - of values with typical log density (near distribution entropy)
  - containing $1 - \epsilon$ of the probability mass
- A typical set A_ϵ **suffices for integration**

$$\int_{\Theta} f(\theta)\, p(\theta|y)\, \mathrm{d}\theta = \int_{A_\epsilon} f(\theta)\, p(\theta|y)\, \mathrm{d}\theta$$

Typical Draws from Multi-Normal

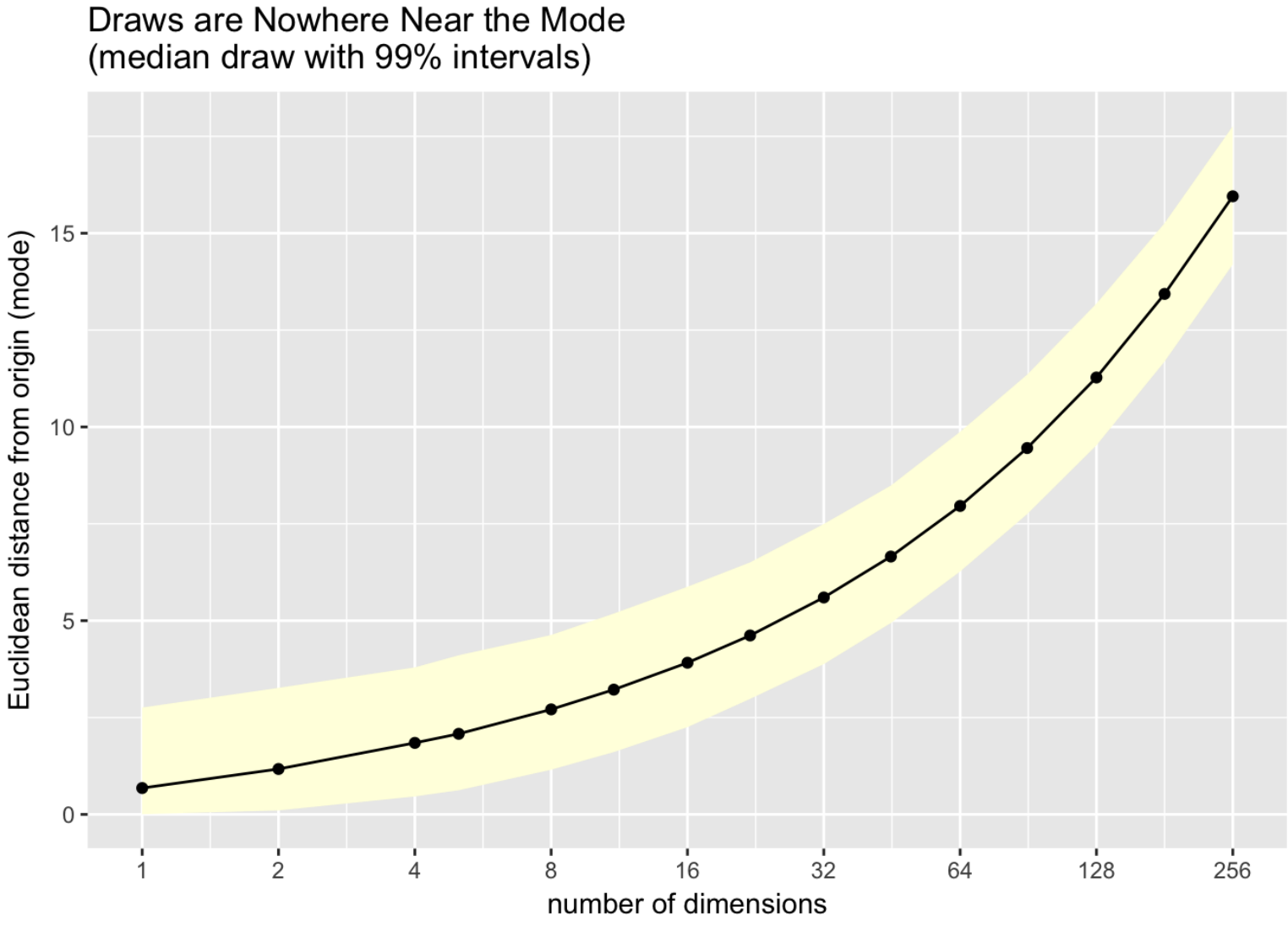
- $Y \sim \text{MultiNormal}(0, \text{I}_N)$ is standard multivariate normal
- $Y_n \sim \text{Normal}(0, 1)$ is thus independently standard normal
- Joint density: $p_Y(y) = \prod_{n=1}^{N} \text{Normal}(y_n \mid 0, 1)$
- Mean, median, and mode (max) of $p_Y(y)$ at $y = 0$

- How far do we expect Y to be from the mode?
- What is the log density of a typical draw of Y?

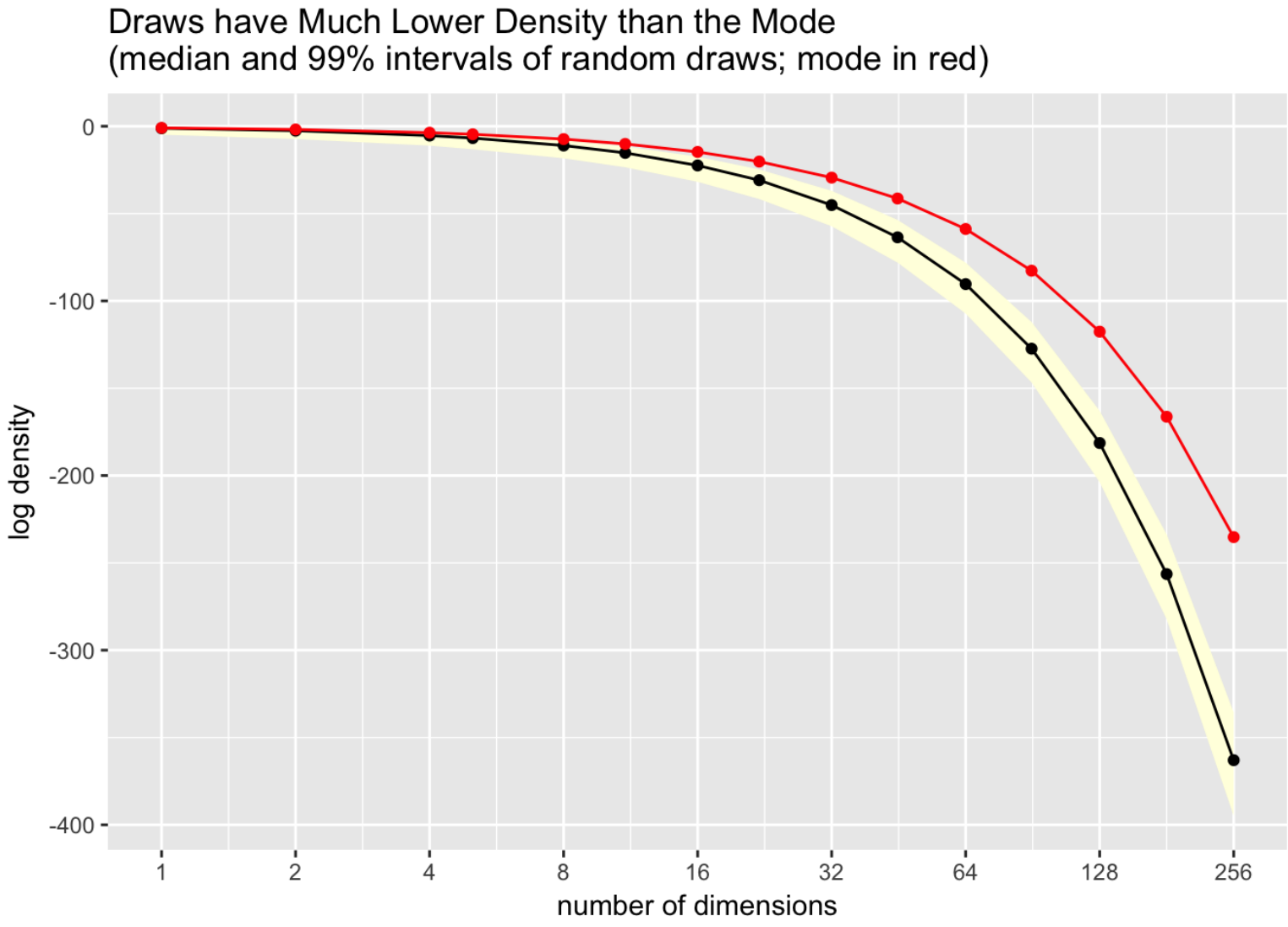
Multi-Normal Draws in Stan

```
generated quantities {
  real dist_to_origin[256];
  real log_lik[256];
  real log_lik_mean[256];
  {
    real sq_dist = 0;  real ll = 0;  real llm = 0;
    for (n in 1:256) {
      real y = normal_rng(0, 1);
      ll = ll + normal_lpdf(y | 0, 1);
      llm = llm + normal_lpdf(0 | 0, 1);
      sq_dist = sq_dist + y^2;
      dist_to_origin[n] = sqrt(sq_dist);
      log_lik[n] = ll;
      log_lik_mean[n] = llm;
    }
  }
}
```

Normal Variate Distance to Mode



Normal Variate Log Density



Normal Mode not in Typical Set

- Plots show that in a standard normal of **more than 5 dimensions**, that the **mode is not in the typical set**
- An **Asimov data set** uses an average member of a set represent the whole set
  - based on Isaac Asimov's short story "Franchise" in which a single average voter represented everyone
  - the average member of a multivariate normal is the mean
  - thus no members of the typical set are average in this sense
  - popular in physics
  - **very poor** solution for most inferential purposes

Concentration of Measure

- We care about probability **mass**, not **density**
- Events with non-zero probability have probability mass, e.g., $\Pr[\theta_0 > \theta_1 \mid y]$
- Mass arises from integrating over density
- As data size increases, posterior concentrates around true value

Sampling Efficiency

- We care only about N_{eff} per second
- Decompose into
    1. Iterations per second
    2. Effective samples per iteration
- Gibbs and Metropolis have high iterations per second (especially Metropolis)
- But they have low effective samples per iteration (especially Metropolis)
- Both are particular weak when there is high correlation among the parameters in the posterior

The Folk Theorem

"When you have computational problems, often there's a problem with your model."

— Andrew Gelman (2008)

- The usual culprits are
  - bugs in: samplers, data munging, model coding, etc.
  - model misspecification

http://andrewgelman.com/2008/05/13/the_folk_theore/

Model Calibration

- Consider 100 days for which a meteorologist predicted a 70% chance of rain
  - about 70 of them should have had rain
  - not fewer, not more!
  - technically, expect Binomial$(100, 0.7)$ rainy day from a calibrated model
- Use posterior predictive checks to test calibration on
  - training data—can it fit?
  - held out data—can it predict?
  - cross-validation—approximates held out with trainin data
- Also applies to interval coverage of parameter values

Model Sharpness

- Ideal forecasts are deterministic
  - predict 100% chance of rain or 0% chance of rain
  - always right
- A forecast of 90% chance of rain reduces uncertainty more than a 50% prediction
- A model is **sharp** if it has narrow posterior intervals
  - Prediction Pr$[\alpha \in (1.2, 1.9)] = 0.9$
  - is sharper than Pr$[\alpha \in (1, 2)] = 0.9$
- I.e., sharper models are more certain in its predictions
- Given calibration, we want our predictions to be **sharp**

Cross-Validation

- Uses single data set to model held-out performance
- Assumes stationarity (as most models do)
- Partition data evenly into disjoint subsets (called **folds**)
  - 10 is a common choice
  - **leave-one-out** (LOO) uses a the number of training data points
- For each fold
  - estimate model on all data but that fold
  - test on that fold
- Usual comparison statistic is held out log likelihood

What Stan Does

Full Bayes: No-U-Turn Sampler

- Adaptive **Hamiltonian Monte Carlo** (HMC)
  - **Potential Energy**: negative log posterior
  - **Kinetic Energy**: random standard normal per iteration
- Adaptation **during warmup**
  - step size adapted to target total acceptance rate
  - mass matrix (scale/rotation) estimated with regularization
- Adaptation **during sampling**
  - simulate forward and backward in time until U-turn
  - **slice sample** along path

(Hoffman and Gelman 2011, 2014)

Euclidean Hamiltonian Monte Carlo

- **Phase space**: q position (parameters); p momentum
- **Posterior density**: $\pi(q)$
- **Mass matrix**: M
- **Potential energy**: $V(q) = -\log \pi(q)$
- **Kinetic energy**: $T(p) = \frac{1}{2} p^\top M^{-1} p$
- **Hamiltonian**: $H(p, q) = V(q) + T(p)$
- **Diff eqs**:

$$\frac{dq}{dt} = +\frac{\partial H}{\partial p} \qquad \frac{dp}{dt} = -\frac{\partial H}{\partial q}$$

Leapfrog Integrator Steps

- Solves Hamilton's equations by **simulating dynamics**
  (symplectic [volume preserving]; ϵ^3 error per step, ϵ^2 total error)
- Given: **step size** ϵ, **mass matrix** M, **parameters** q
- **Initialize kinetic** energy, $p \sim \text{Normal}(0, \mathbf{I})$
- **Repeat** for L leapfrog steps:

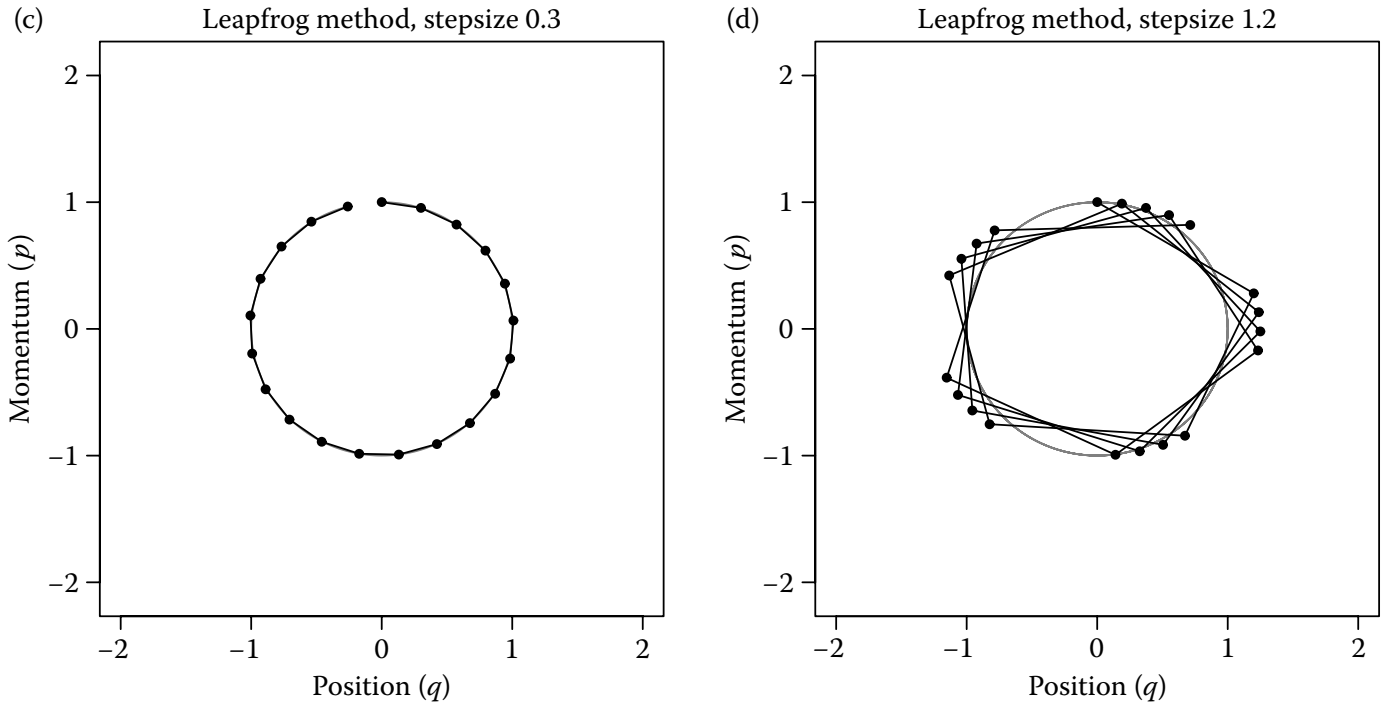
$$p \leftarrow p - \frac{\epsilon}{2} \frac{\partial V(q)}{\partial q} \qquad \text{[half step in momentum]}$$

$$q \leftarrow q + \epsilon M^{-1} p \qquad \text{[full step in position]}$$

$$p \leftarrow p - \frac{\epsilon}{2} \frac{\partial V(q)}{\partial q} \qquad \text{[half step in momentum]}$$

Leapfrog Algorithm Example

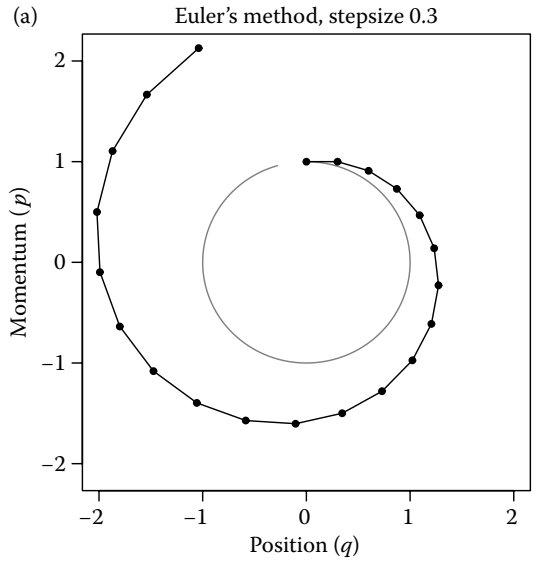
- Leapfrog algorithm for Hamiltonian dynamics (1 param)
- Position vs. momentum (phase space)



Radford Neal. MCMC using Hamiltonian dynamics. In *Handbook of MCMC*.

Numerical Divergences

- Hamiltonian should be conserved; sometimes it isn't
- If it goes too far, we say it has "diverged"
  - Here's an example with Euler's method (not HMC)



Radford Neal. MCMC using Hamiltonian dynamics. In *Handbook of MCMC*.

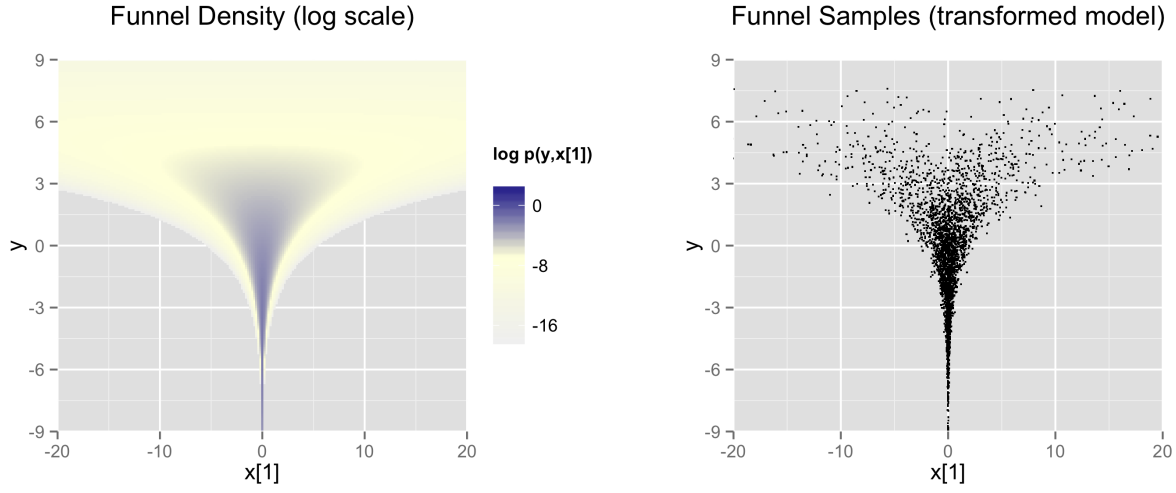
The Funnel

Position-Dependent Curvature

- Mass matrix does **global** adaptation for
  - parameter scale (diagonal) and rotation (dense)
- Dense mass matrices hard to estimate ($\mathcal{O}(N^2)$ estimands)
- **Problem**: Position-dependent curvature
  - Example: banana-shaped densities
    - arise when parameter is product of other parameters
  - Example: hierarchical models
    - hierarchical variance controls lower-level parameters
- Mitigate by reducing stepsize
  - initial (`stepsize`) and target acceptance (`adapt_delta`)

Funnel-Shaped Posteriors (1/2)

- Arise in hierarchical model with no data (Neal 2003)
  - $\log \sigma \sim \text{Normal}(0, 1.5)$ [hierarchical scale]
  - $\beta_n \sim \text{Normal}(0, \sigma)$ for $n \in 1:9$ [low-level coefficients]



β_1 coefficient (x-axis) vs. $\log \sigma$ (y-axis); *left*) density plot (log scale); *right*) 4000 independent draws

Neal, R. M. 2003. Slice sampling. *Annals of Statistics*, **31**(3):705–767

Funnel-Shaped Posteriors (2/2)

- Very **challenging for sampling**
- Need large step size to explore mouth of funnel
- Need small step size to explore neck of funnel
- Even small step sizes lead to divergences
  - numerical failure of Hamiltonian dynamics simulation to conserve the Hamiltonian
- Betancourt and Girolami (2015) analyzed for Hamiltonian Monte Carlo

Betancourt and Girolami. 2015. Hamiltonian Monte Carlo for hierarchical models.

In *Current Trends in Bayesian Methodology with Applications*. CRC

Non-Centered Parameterization

- The non-centered parameterization of the funnel is

$$\begin{aligned} \log \sigma &\sim \text{Normal}(0, 1.5) \\ \beta_n^{\text{std}} &\sim \text{Normal}(0, 1) \\ \beta_n &= \sigma \times \beta_n^{\text{std}} \end{aligned}$$

- Removes dependency of β on σ in prior
- Called it “Matt trick” (after Matt Hoffman) before realizing it was well-known

Adding Data

- Use the β in an intercept-only binomial logistic regression,

$$y_j \sim \text{Binomial}(N, \text{logit}^{-1}(\beta_j))$$

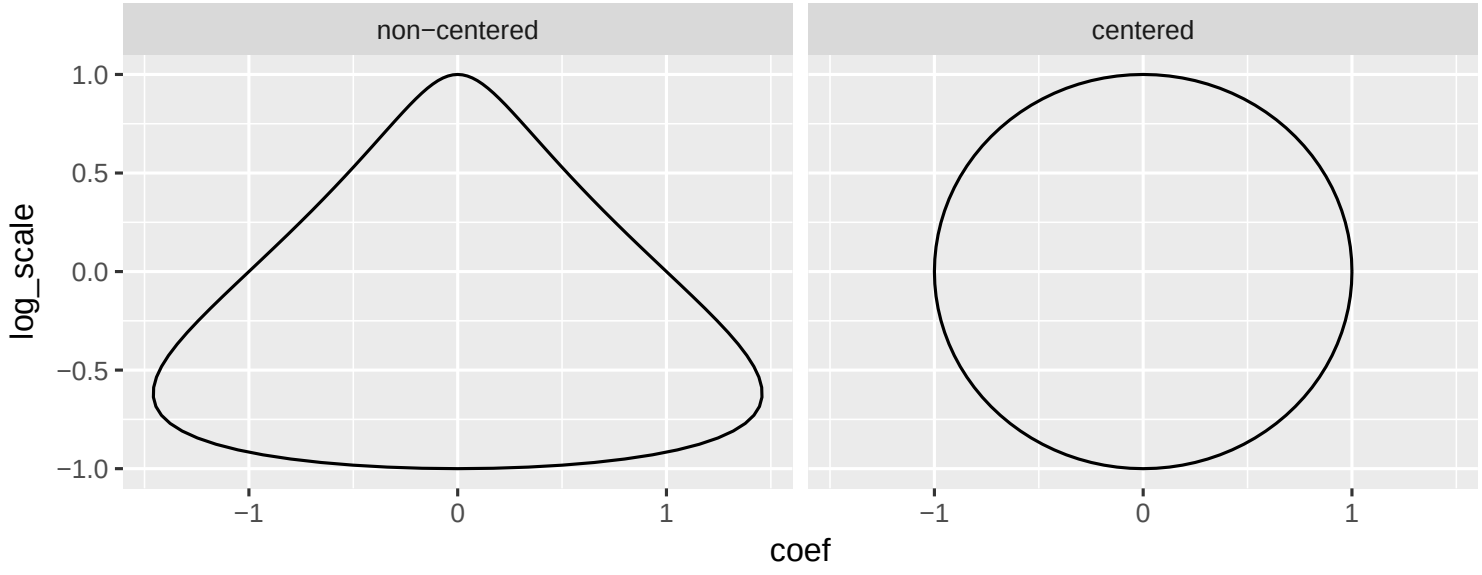
  - i.e., $y_j \in 0:N$ is number of successes in K trials
  - β_n is log odds of success for group j

- More data lessens dependency between β and σ

- With informative enough data, centered parameterization is better
  - not size of data, but how much it constrains posterior

Non-Centered + Data = Funnel

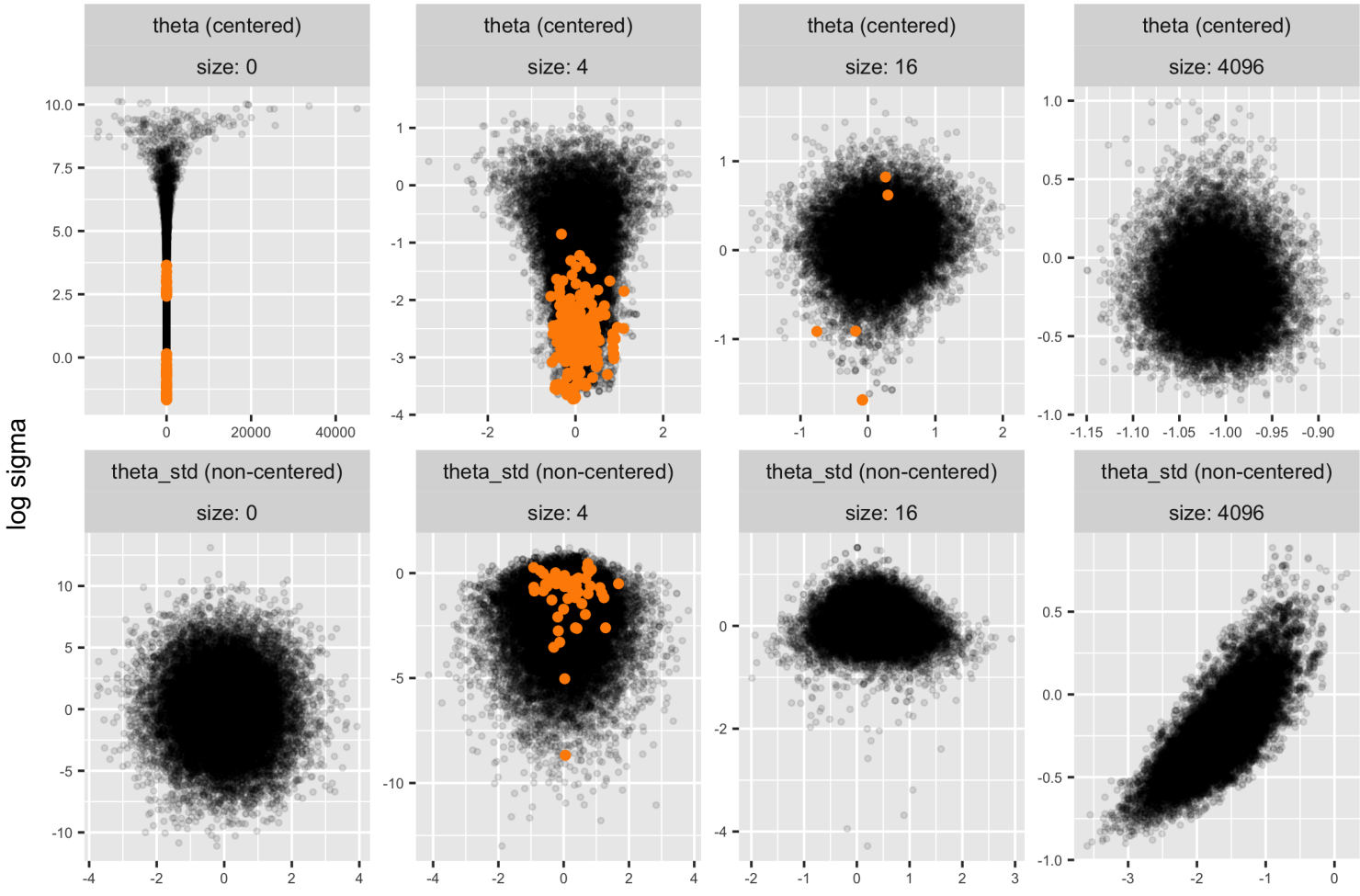
- With more data, centered approaches independent normal
- Non-centering ($\beta^{\text{std}} = \beta / \sigma$) produces a funnel
- Centered parameterizations dominate with lots of data



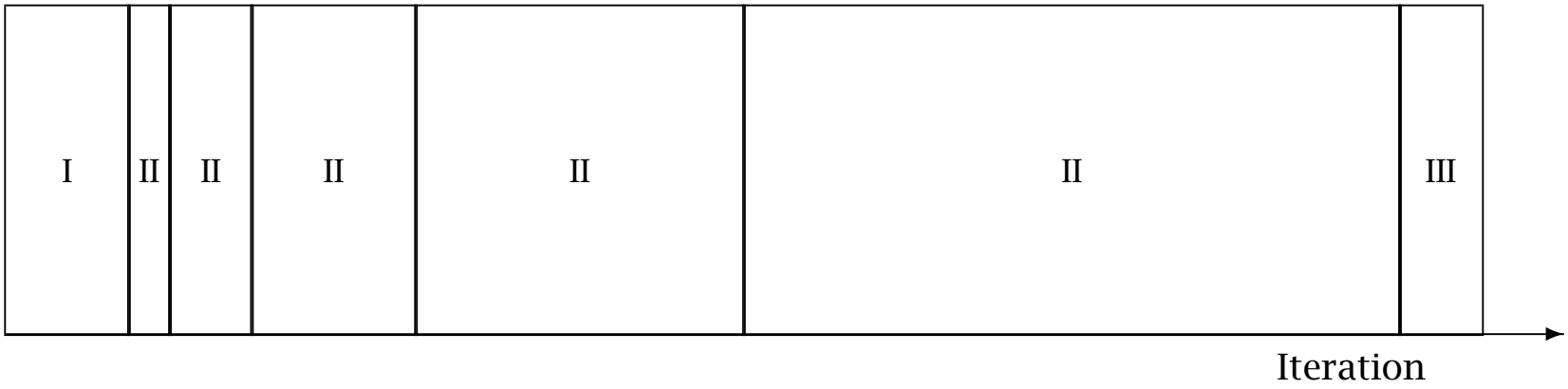
Funnel + Data in the Wild

centered vs. non-centered parameterization

hierarchical logistic regression with 10 groups, intercept only, 20000 draws, divergences in orange



Adaptation During Warmup



- (I) initial fast interval to find typical set
  (step size, default 75 iterations)
- (II) expanding memoryless windows to estimate metric
  (step size & metric, initial 25 iterationss)
- (III) final fast interval (step size, default 50 its)

Identification

Identifiability

- Notion from classical maximum likelihood estimation
- Roughly, a model is **non-identifiable** if
  - there do not exist parameters $\theta \neq \theta'$
  - such that for all data sets y
  - $p(y|\theta) = p(y'|theta)$

Overparameterized Normal

- Consider the difference between
  - Standard Parameterization

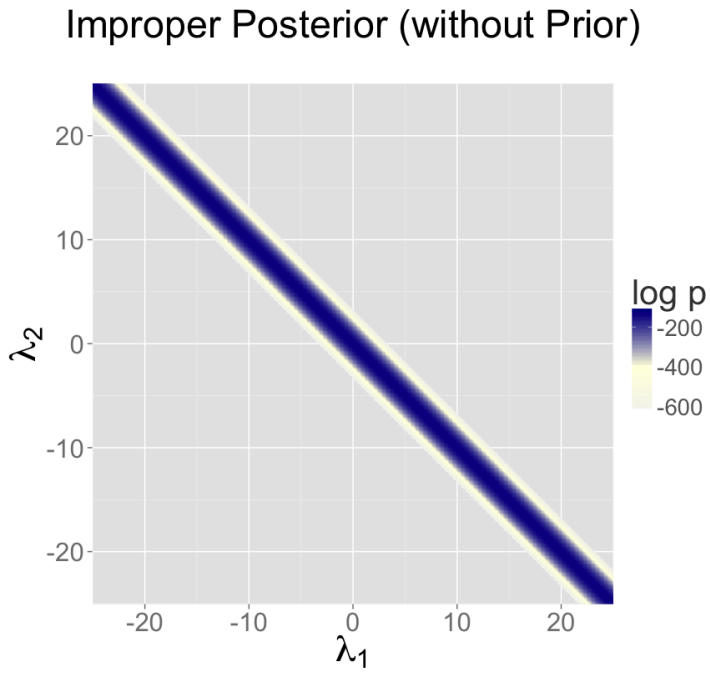
$$y \sim \text{Normal}(\mu, \sigma)$$

  - Overparameterization

$$y \sim \text{Normal}(\lambda_1 + \lambda_2, \sigma)$$

- What's going to happen?

Fit of Overparameterized Normal



- Characteristic ridge of probability mass
- Imrpoper (i.e., can't be normalized)

Identifiability

- Notion from classical maximum likelihood estimation
- Roughly, a model is **non-identifiable** if
  - there do not exist parameters $\theta \neq \theta'$
  - such that for all data sets y
  - $p(y|\theta) = p(y'|\theta)$
- Overparameterized normal not identified; for all y,

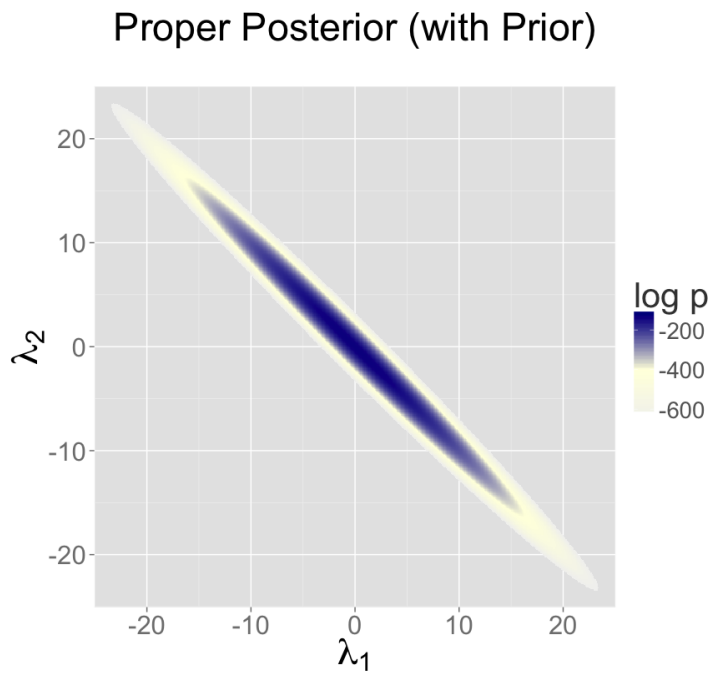
$$\text{Normal}(y|1 + -1, \sigma) \;=\; \text{Normal}(y|2 + -2, \sigma).$$

Soft Identification by Prior

- Mitigate problem by adding prior, e.g.,

$$\lambda_1, \lambda_2 \sim \text{Normal}(0, 10)$$

- Proper prior induces proper posterior



Effectiveness

- $N = 100$ data points, four chains with 2000 iterations, half warmup
- Two parameter model, not identified

	Mean	MCSE	StdDev	N_Eff	R_hat
lambda1	1.3e+03	1.9e+03	2.7e+03	2.1	5.2
lambda2	-1.3e+03	1.9e+03	2.7e+03	2.1	5.2
sigma	1.0e+00	8.5e-03	6.2e-02	54	1.1

- Two parameter model, soft identification with prior

	Mean	MCSE	StdDev	N_Eff	R_hat
lambda1	0.31	2.8e-01	7.1e+00	638	1.0
lambda2	-0.14	2.8e-01	7.1e+00	638	1.0
sigma	1.0	2.6e-03	8.0e-02	939	1.0

Hard Identification

- Even better to just use identifiable model

$$y \sim \text{Normal}(\mu, \sigma)$$

- One parameter model, identified

	Mean	MCSE	StdDev	N_Eff	R_hat
mu	0.17	2.1e-03	0.10	2408	1.0
sigma	1.0	1.6e-03	0.071	2094	1.0