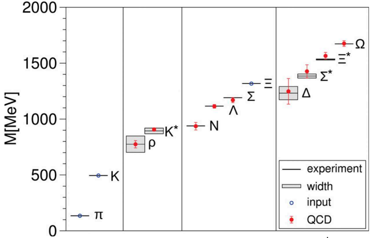
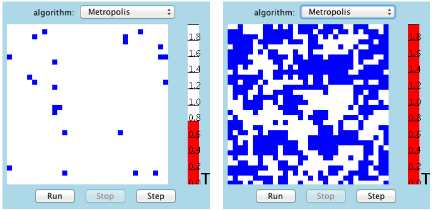
Lattice QCD
light hadron mass spectrum



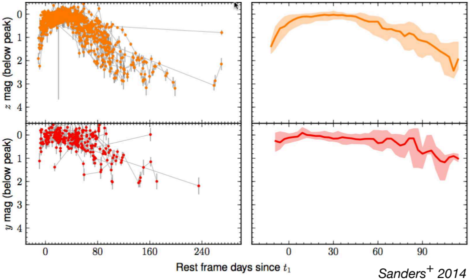
Dürr+ 2008

Ising model simulations



TU Delft applet

Pan-STARRS1 SN IIP light curves



Sanders+ 2014

Lattice QCD: Sample matter (ψ) and gauge (U) fields $\propto \exp(-S[U, \bar{\psi}, \psi])$

Ising model: Sample matter configurations $\propto \exp(-H/T)$

Bayesian inference: Sample parameters $\propto \pi(\theta)\mathcal{L}(\theta)$

All involve sampling from high-dimensional, dependent probability distributions

$\Rightarrow$ *How can we build high-dimensional pseudo-random number generators?*

Hamiltonian Monte Carlo: Recent Developments

Tom Loredo
Dept. of Astronomy, Cornell University

Astrophysics Lunch — 15 Oct 2014

Agenda

1. **Monte Carlo integration/posterior sampling**
2. **Hamiltonian Monte Carlo**
3. **Challenges, developments**
4. **Stan**

Notation

$$
\begin{aligned}
p(\theta|D, M) &= \frac{p(\theta|M)p(D|\theta, M)}{p(D|M)} \\
&= \frac{\pi(\theta)\mathcal{L}(\theta)}{Z} = \frac{q(\theta)}{Z}
\end{aligned}
$$

- M = model specification
- D specifies observed data
- θ = model parameters
- $\pi(\theta)$ = prior pdf for θ
- $\mathcal{L}(\theta)$ = likelihood for θ (likelihood function)
- $q(\theta) = \pi(\theta)\mathcal{L}(\theta) =$ "quasiposterior"
- $Z = p(D|M) =$ (marginal) likelihood for the model

Marginal likelihood:

$$Z = \int d\theta \; \pi(\theta)\mathcal{L}(\theta) = \int d\theta \; q(\theta)$$

Statistical mechanics analogy:

$$q(\theta) = \exp[-U(\theta)/T] \quad \text{for} \quad U(\theta) \equiv -\log[\pi(\theta)\mathcal{L}(\theta)]$$

Posterior corresponds to $T = 1$
Marginal likelihood is $Z(1)$ for partition function $Z(T)$

Bayesian Computation

Parameter space integrals

For model with m parameters, we need to evaluate integrals like:

$$\int d^m\theta \, g(\theta) \, \pi(\theta) \, \mathcal{L}(\theta) \quad = \quad \int d^m\theta \, g(\theta) \, q(\theta) \quad \pi(\theta) \, \mathcal{L}(\theta)$$

- $g(\theta) = 1 \rightarrow p(D|M)$ (norm. const., model likelihood)
- $g(\theta) = \theta \rightarrow$ posterior mean for θ
- $g(\theta) =$ 'box' $\rightarrow$ probability $\theta \in$ credible region
- $g(\theta) = 1$, integrate over subspace $\rightarrow$ marginal posterior
- $g(\theta) = \delta[\psi - \psi(\theta)] \rightarrow$ propagate uncertainty to $\psi(\theta)$

Monte Carlo Integration

$\int g \times p$ is just the *expectation of g*; suggests approximating with a *sample average*:

$$\int d\theta\, g(\theta)p(\theta) \approx \frac{1}{n} \sum_{\theta_i \sim p(\theta)} g(\theta_i) + O(n^{-1/2}) \quad \left[\begin{array}{c} \sim O(n^{-1}) \text{ with} \\ \text{quasi-MC} \end{array} \right]$$

This is like a cubature rule, with *equal weights* and *random nodes*

Ignores smoothness $\rightarrow$ poor performance in 1-D, 2-D

Avoids curse: $O(n^{-1/2})$ regardless of dimension

Why/when it works

- Independent sampling & law of large numbers $\rightarrow$ asymptotic convergence in probability
- Error term is from CLT; requires finite variance

Practical problems

- $p(\theta)$ must be a density we can draw IID samples from—perhaps the prior, but...
- $O(n^{-1/2})$ multiplier (std. dev'n of g) may be large

$\rightarrow$ *IID* Monte Carlo can be hard if dimension $\gtrsim$ 5–10*

*IID = independently, identically distributed

Posterior sampling

$$\int d\theta\ g(\theta)p(\theta|D) \approx \frac{1}{n} \sum_{\theta_i \sim p(\theta|D)} g(\theta_i) + O(n^{-1/2})$$

When $p(\theta)$ is a posterior distribution, drawing samples from it is called *posterior sampling*:

- *One set of samples* can be used for many different calculations (so long as they don't depend on low-probability events)
- This is the most promising and general approach for Bayesian computation in *high dimensions*—though with a twist (MCMC!)

Challenge: How to build a RNG that samples from a posterior?

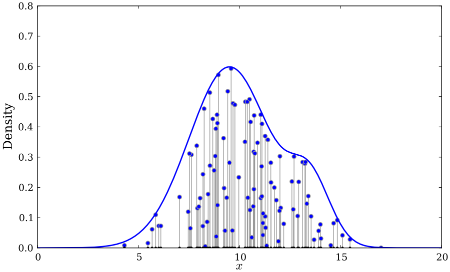
Accept-Reject Algorithm

Goal: Given $q(\theta) \equiv \pi(\theta)\mathcal{L}(\theta)$, build a RNG that draws samples from the probability density function (pdf)

$$f(\theta) = \frac{q(\theta)}{Z} \quad \text{with} \quad Z = \int d\theta\, q(\theta)$$

The probability for a region under the pdf is the *area (volume) under the curve (surface)*.

$\rightarrow$ Sample points uniformly in volume under q; their θ values will be draws from $f(\theta)$.



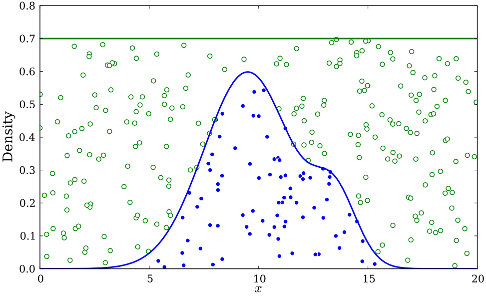
The fraction of samples with θ ("x" in the fig) in a bin of size $\delta\theta$ is the fractional area of the bin.

How can we generate points uniformly under the pdf?

Suppose $q(\theta)$ has compact support: it is nonzero over a finite contiguous region of θ-space of length/area/volume V.

Generate *candidate* points uniformly in a rectangle enclosing $q(\theta)$.

Keep the points that end up under q.



Basic accept-reject algorithm

1. Find an upper bound Q for $q(\theta)$
2. Draw a candidate parameter value θ' from the uniform distribution in V
3. Draw a uniform random number, u
4. If the ordinate $uQ < q(\theta')$, record θ' as a sample
5. Goto 2, repeating as necessary to get the desired number of samples.

Efficiency $=$ ratio of areas (volumes), $Z/(QV)$.

Curse of dimensionality: Efficiency declines *quickly* with dimension!

Take-away idea: *Propose candidates that may be accepted or rejected*

Markov Chain Monte Carlo

Accept/Reject aims to produce *independent* samples—each new θ is chosen irrespective of previous draws.

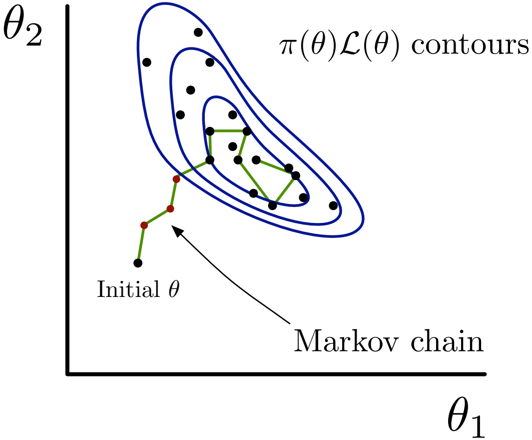
To enable exploration of complex pdfs, let's introduce *dependence*: Choose new θ points in a way that

- Tends to *move toward* regions with higher probability than current
- Tends to *avoid* lower probability regions

The simplest possibility is a *Markov chain*:

$$\begin{aligned} p(\text{next location}|\textit{current}\ \textbf{and}\ \textit{previous locations}) \\ = T(\text{next location}|\textit{current location}) \end{aligned}$$

A Markov chain "has no memory."


θ_2
$\pi(\theta)\mathcal{L}(\theta)$ contours
Initial θ
Markov chain
θ_1

Reversibility/Detailed Balance

A sufficient (but not necessary!) condition for there to be an equilibrium distribution is the *detailed balance* or *reversibility* condition:

$$\begin{aligned} p_{\mathrm{eq}}(x)T(y|x) &= p_{\mathrm{eq}}(y)T(x|y) \qquad \textit{or} \\ \frac{T(y|x)}{T(x|y)} &= \frac{p_{\mathrm{eq}}(y)}{p_{\mathrm{eq}}(x)} \end{aligned}$$

If we set $p_{\mathrm{eq}} = q/Z$, and we build a reversible transition distribution for this choice, then *the equilibrim distribution will be the posterior distribution*

How can we build $T(y|x)$ to target a particular $p_{\mathrm{eq}}(x)$?

Metropolis-Hastings algorithm

Given a target quasi-distribution $q(x)$ (it need not be normalized):

1. Specify a proposal distribution $k(y|x)$ (make sure it is irreducible and aperiodic).
2. Choose a starting point x; set $t = 0$ and $S_t = x$
3. Increment t
4. Propose a new state $y \sim k(y|x)$
5. If $q(x)k(y|x) < q(y)k(x|y)$, set $S_t = y$; goto (3)
6. Draw a uniform random number u
7. If $u < \frac{q(y)k(x|y)}{q(x)k(y|x)}$, set $S_t = y$; else set $S_t = x$; goto (3)

The art of MCMC is in *specifying the proposal distribution $k(y|x)$*

We want:

- New proposals to be accepted, so there is movement
- Movement to be significant, so we explore efficiently

These desiderata compete!

Random walk Metropolis (RWM)

Propose an *increment*, z, from the current location, not dependent on the current location, so $y = x + z$ with a specified PDF $K(z)$, corresponding to

$$k(y|x) = K(y - x)$$

The proposals would give rise to a *random walk* if they were all accepted; the M-H rule modifies them to be a kind of directed random walk

Most commonly, a symmetric proposal is adopted:

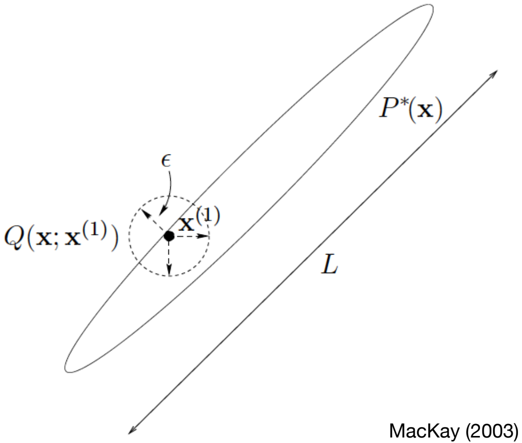
$$k(y|x) = K(|y - x|)$$

The acceptance probability simplifies:

$$\alpha(y|x) = \min\left[\frac{q(y)}{q(x)}, 1\right]$$

Key issues: shape and scale (in all directions) of $K(z)$

RWM in 2-D



Small step size $\rightarrow$ good acceptance rate, but slow exploration

Random Walks

Random walk Metropolis and most other MCMC updates execute a *random walk* through parameter space:

- Moves are local, with a characteristic scale l
- Total distance traversed over time $t \propto \sqrt{t}$

This is a relatively slow (albeit steady) rate of exploration

Multimodality $\rightarrow$ even slower exploration; only rare large jumps can move between modes

We need methods designed to make large moves

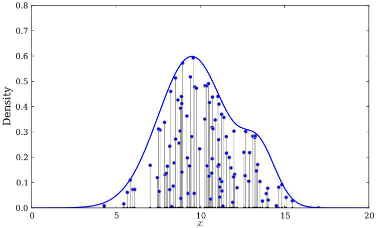
Agenda

1. Monte Carlo integration/posterior sampling
2. **Hamiltonian Monte Carlo**
3. Challenges, developments
4. Stan

Auxiliary variables

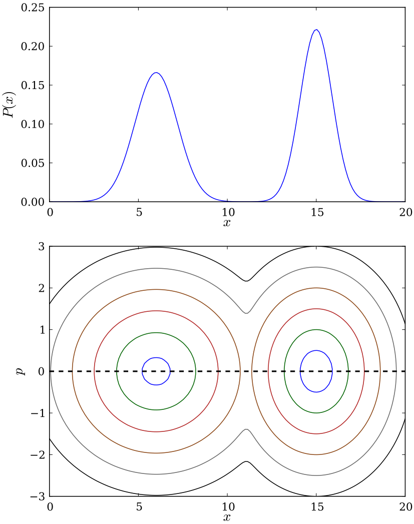
The accept/reject method for sampling a d-D density:

- Sample from a *uniform* $(d+1)$-D density (with a complicated boundary):



- Report the marginal samples for the d original dimensions

A paradoxical notion motivating some advanced MCMC methods is that making the problem "harder" (higher-dimensional) may actually make it *easier*



Double the dimensionality!

$$p(x, P) \propto q(x) \times f(P)$$

$$p(x) = \int dP\, p(x, P) \propto q(x)$$

$$p(P) = \int dx\, p(x, P) \propto f(P)$$

- Pick $P \sim f(P)$
- Move along a contour in phase space
- Drop P, keep x

Will work if the phase space motion corresponds to sampling $p(x, P)$

Hamiltonian (Hybrid) Monte Carlo

Give samples "momentum" so moves tend to go in the same direction a while; use derivatives to guide the evolution $\rightarrow$ suppress random walks

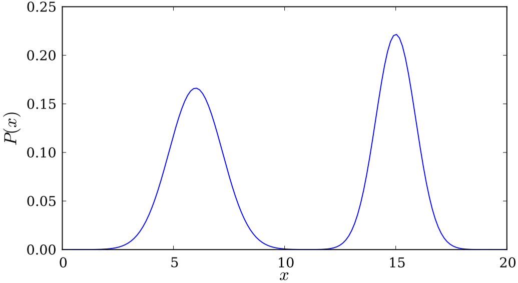
Adds d additional variables, P, with a joint Gaussian dist'n:

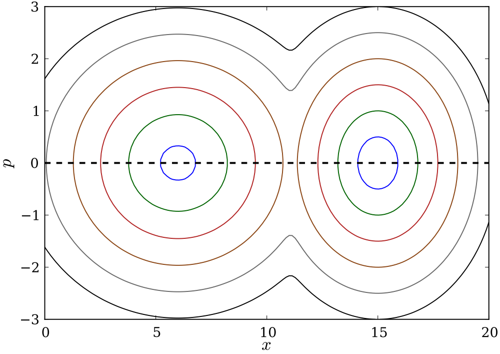
$$\log p(\theta, P) = -\left[U(\theta) + \frac{1}{2}P^2\right]; \qquad U(\theta) \equiv -\log q(\theta)$$

Sample P from a Gaussian, and use it to generate proposals via

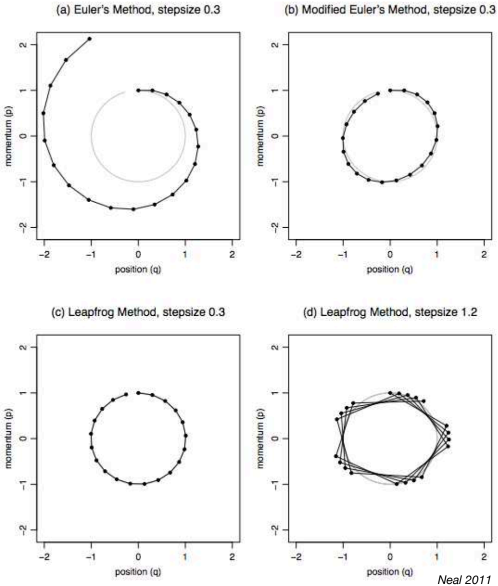
$$\dot{\theta} = P; \qquad \dot{P} = -\frac{\partial H}{\partial \theta}$$

Hamiltonian dynamics $\rightarrow$ reversible, preserves volume, keeps p constant (proposals always accepted)


0.25
0.20
0.15
0.10
0.05
0.00
$P(x)$
0
5
10
15
20
x

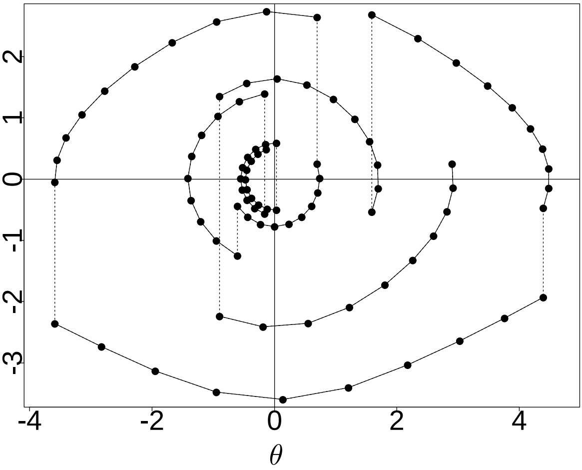

3
2
1
0
−1
−2
−3
p
0
5
10
15
20
x

Numerical integration (1-D)



Neal 2011

Sampling a 1-D Student-t dist'n with dof$= 5$



HMC vs. random walk (2-D)

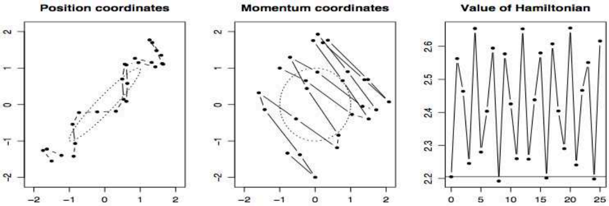
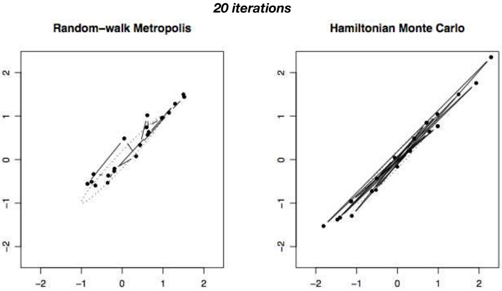


Figure 3: A trajectory for a 2D Gaussian distribution, simulated using 25 leapfrog steps with a stepsize of 0.25. The ellipses plotted are one standard deviation from the means. The initial state had $q = [-1.50, -1.55]^T$ and $p = [-1, 1]^T$.



Neal 2011

Agenda

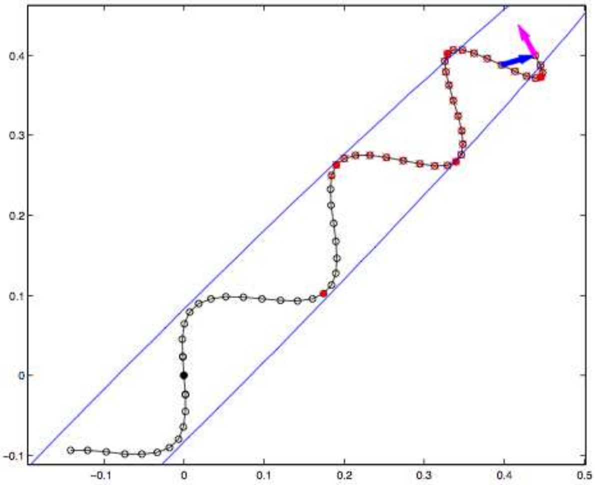
1. Monte Carlo integration/posterior sampling
2. Hamiltonian Monte Carlo
3. **Challenges, developments**
4. Stan

Challenges for basic HMC

- Tuning parameters:
  - Choosing time step size, ϵ, and integration length, L
  - Handling problems with very different scales along different dimensions
- Computing the needed derivatives

Tuning integration length

No-U-Turn Sampler (NUTS)

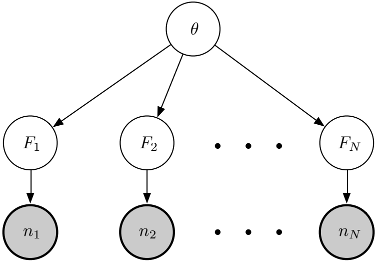


Hoffman & Gelman 2013

Multilevel models: parameter-dependent scales

Goal: Learn a flux dist'n from photon counts

Qualitative



Population parameters

Source properties

Observed data

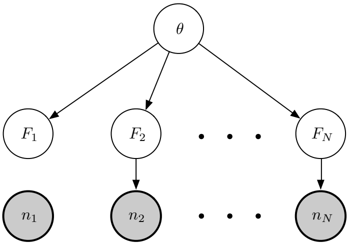
Quantitative

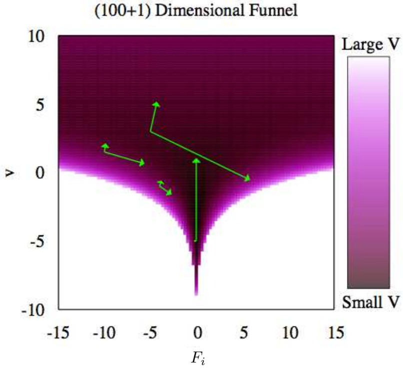
$$\theta = (\alpha, s) \text{ or } (\mu, \sigma)$$

$$\pi(\theta) = \mathrm{Flat}(\mu, \sigma)$$

$$p(F_i|\theta) = \mathrm{Gamma}(F_i|\theta)$$

$$p(n_i|F_i) = \mathrm{Pois}(n_i|\epsilon_i F_i)$$





Betancourt & Girolami 2013

Mass matrix = metric

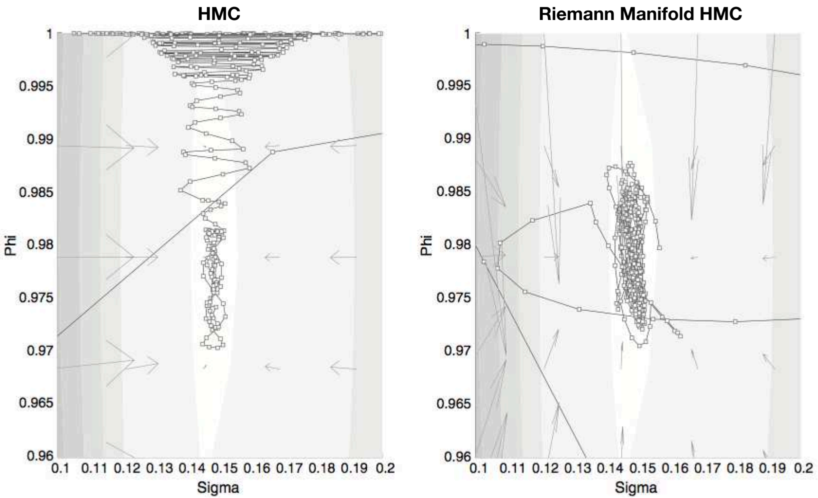
Add d additional variables, P, with a *correlated* Gaussian dist'n:

$$\log p(\theta, P) = -\left[U(\theta) + \frac{1}{2} P \cdot M^{-1} \cdot P\right]; \qquad U(\theta) \equiv -\log p(\theta)$$

M introduces d more tuning parameters!

- **Euclidean manifold HMC:** Use the Hessian at the mode
- **Riemannian manifold HMC:** Use position-dependent $M(\theta)$

Riemann manifold HMC



Girolami & Calderhead 2011

Agenda

1. Monte Carlo integration/posterior sampling

2. Hamiltonian Monte Carlo

3. Challenges, developments

4. **Stan**

Stan: mc-stan.org

Stan

Stan is a probabilistic programming language implementing full Bayesian statistical inference with

- MCMC sampling (NUTS, HMC)

and penalized maximum likelihood estimation with

- Optimization (BFGS)

Stan is coded in C++ and runs on all major platforms (Linux, Mac, Windows).

Stan is freedom-respecting, open-source software (new BSD core, GPLv3 interfaces).

Interfaces

Download and getting started instructions, organized by interface:

- RStan v2.4.0 (R)
- PyStan v2.4.0 (Python)
- CmdStan v2.4.0 (shell, command-line terminal)

Manual & Examples

Models are portable across interfaces, so these are cross-platform:

- Modeling Language Manual
- Example Models

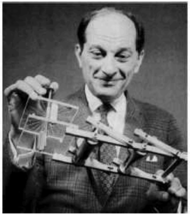
Home | RStan | PyStan | CmdStan | Manual | Examples | Groups | Issues | Contribute | Source | Citations | Team | Shop

http://mc-stan.org/

https://groups.google.com/d/forum/stan-users

How Stan Got its Name

- “Stan” is *not* an acronym; Gelman mashed up
    1. Eminem song about a stalker fan, and
    2. Stanislaw Ulam (1909–1984), co-inventor of Monte Carlo method (and hydrogen bomb).



Ulam holding the Fermiac, Enrico Fermi’s physical Monte Carlo simulator for random neutron diffusion

From Daniel Lee

Stan capabilities

- Hamiltonian Monte Carlo (HMC)
  - sample parameters on unconstrained space
    → transform + Jacobian adjustment
  - gradients of the model wrt parameters
    → automatic differentiation
  - sensitive to tuning parameters → **No-U-Turn Sampler**
- No-U-Turn Sampler (NUTS)
  - warmup: estimates mass matrix and step size
  - sampling: adapts number of steps
  - **maintains detailed balance**
- Optimization
  - BFGS, Newton's method

From Daniel Lee

RMHMC, ensemble samplers in progress. . .

Stan "Pumps" example (number counts!)

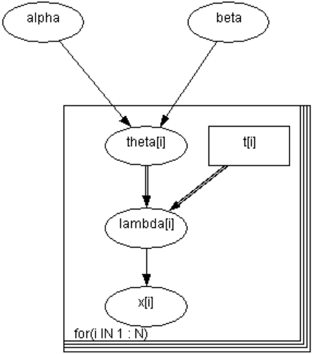
Power law slope

$$\text{Flux } \theta_i \sim \text{Gamma}(\alpha, \beta)$$

Exponential cutoff

$$\text{Expected counts } \lambda_i = \theta_i t_i$$

$$\text{Observed counts } x_i \sim \text{Poisson}(\lambda_i)$$



22 lines (18 sloc) | 0.313 kb

```

data {
  int<lower=0> N;
  int<lower=0> x[N];
  real  t[N];
}

parameters {
  real<lower=0> alpha;
  real<lower=0> beta;
  real<lower=0> theta[N];
}

model {
  alpha ~ exponential(1.0);
  beta ~ gamma(0.1, 1.0);
  for (i in 1:N){
    theta[i] ~ gamma(alpha, beta);
    x[i] ~ poisson(theta[i] * t[i]);
  }
}
```

Inaugural "Stan model of the week"

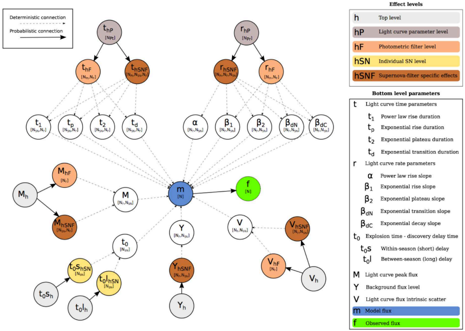
arXiv.org > astro-ph > arXiv:1404.3619

Search

Astrophysics > Instrumentation and Methods for Astrophysics

Unsupervised Transient Light Curve Analysis Via Hierarchical Bayesian Inference

Nathan Sanders (CfA), Michael Betancourt (University of Warwick), Alicia Soderberg (CfA)



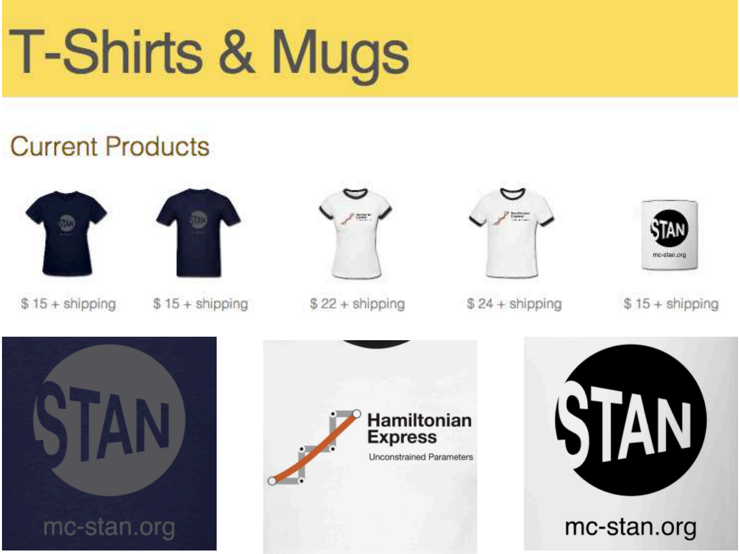
Models light curves of 20,000 Pan-STARRS1 observations of 80 SN IIP

Stan status

- Team: ~12 members, distributed
- 4 Interfaces: CmdStan, RStan, PyStan, MStan
- 700+ on stan-users mailing list
- Actual number of users unknown
  - User manual: 6658 downloads since 2/14
  - PyStan: 1299 downloads in the last month
  - CmdStan / RStan / MStan: ?
- 75+ citations over 2 years
  - stats, astrophysics, political science
  - ecological forecasting: phycology, fishery
  - genetics, medical informatics

From Daniel Lee

Stan Store


T-Shirts & Mugs
Current Products
$ 15 + shipping
$ 15 + shipping
$ 22 + shipping
$ 24 + shipping
$ 15 + shipping
STAN
mc-stan.org
Hamiltonian
Express
Unconstrained Parameters
STAN
mc-stan.org