Posterior Sampling and Posterior Predictive Checking

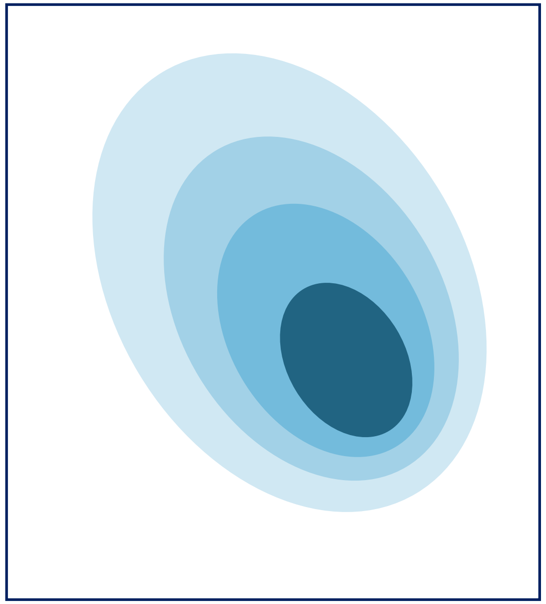
June 11, 2025

UTSSRP

Prof. Gwendolyn Eadie

Sampling from the posterior distribution (Ch 3, McElreath)

- Our example with m&m's was nice and analytic
  - We found an expression for $p(\theta|y)$ no problem
- In many contexts, the posterior distribution $p(\theta|y)$ is intractable, or computationally expensive to calculate directly
- One approach is grid-approximation
- **Better yet, *draw samples* from a distribution proportional to the posterior**
- Once you have samples, it's easy to calculate many quantities of interest
  - Summary statistics
  - Estimating the probability within some interval (e.g., credible intervals)
  - Quantities of scientific interest that depend on the parameters



Bayesian Computation: sampling from a distribution

- Grid approximation
  - Computationally expensive as number of parameters increases
- Markov Chain Monte Carlo (MCMC)
  - **Metropolis algorithm**
    - Invented by physicists
    - Assumes some symmetry in the sampling mechanism
  - Metropolis-Hastings
    - Generalizes to include asymmetry
  - Gibbs sampling
    - Uses conditional probabilities
- Hamilton Monte Carlo
    - Uses physics ideas
    - Will cover this more after reading break

Why posterior samples are so useful

- Model design
  - Draw samples from the prior distribution to get a sense of what the model expects before you use the data (helpful in multidimensional problems)
- Model checking
  - Posterior predictive checks!
- Software validation
  - Simulate data from known model
  - Fit data to model and make sure you get the expected result
- Research design
  - With simulated observations, you ca test whether research design will be effective
- Forecasting
  - Simulate new predictions
  - What modifications to model may be needed in the future
- Scientific inference
  - Propagate uncertainty in future calculations

Metropolis algorithm

Invented by physicists!

Cited over 47,000 times

THE JOURNAL OF CHEMICAL PHYSICS VOLUME 21, NUMBER 6 JUNE, 1953

Equation of State Calculations by Fast Computing Machines

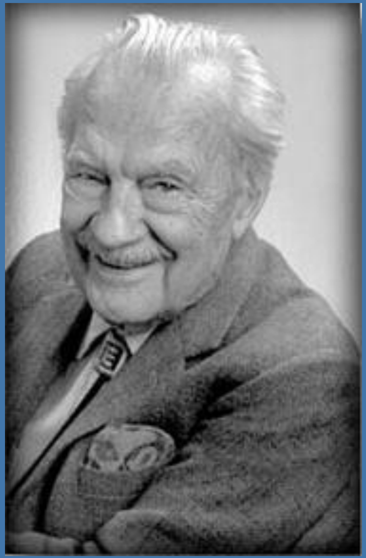
NICHOLAS METROPOLIS, ARIANNA W. ROSENBLUTH, MARSHALL N. ROSENBLUTH, AND AUGUSTA H. TELLER,
Los Alamos Scientific Laboratory, Los Alamos, New Mexico

AND

EDWARD TELLER,* *Department of Physics, University of Chicago, Chicago, Illinois*

(Received March 6, 1953)

A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional rigid-sphere system have been obtained on the Los Alamos MANIAC and are presented here. These results are compared to the free volume equation of state and to a four-term virial coefficient expansion.



THE JOURNAL OF CHEMICAL PHYSICS VOLUME 21, NUMBER 6 JUNE, 1953

Equation of State Calculations by Fast Computing Machines

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, and Augusta H. Teller,
Los Alamos Scientific Laboratory, Los Alamos, New Mexico

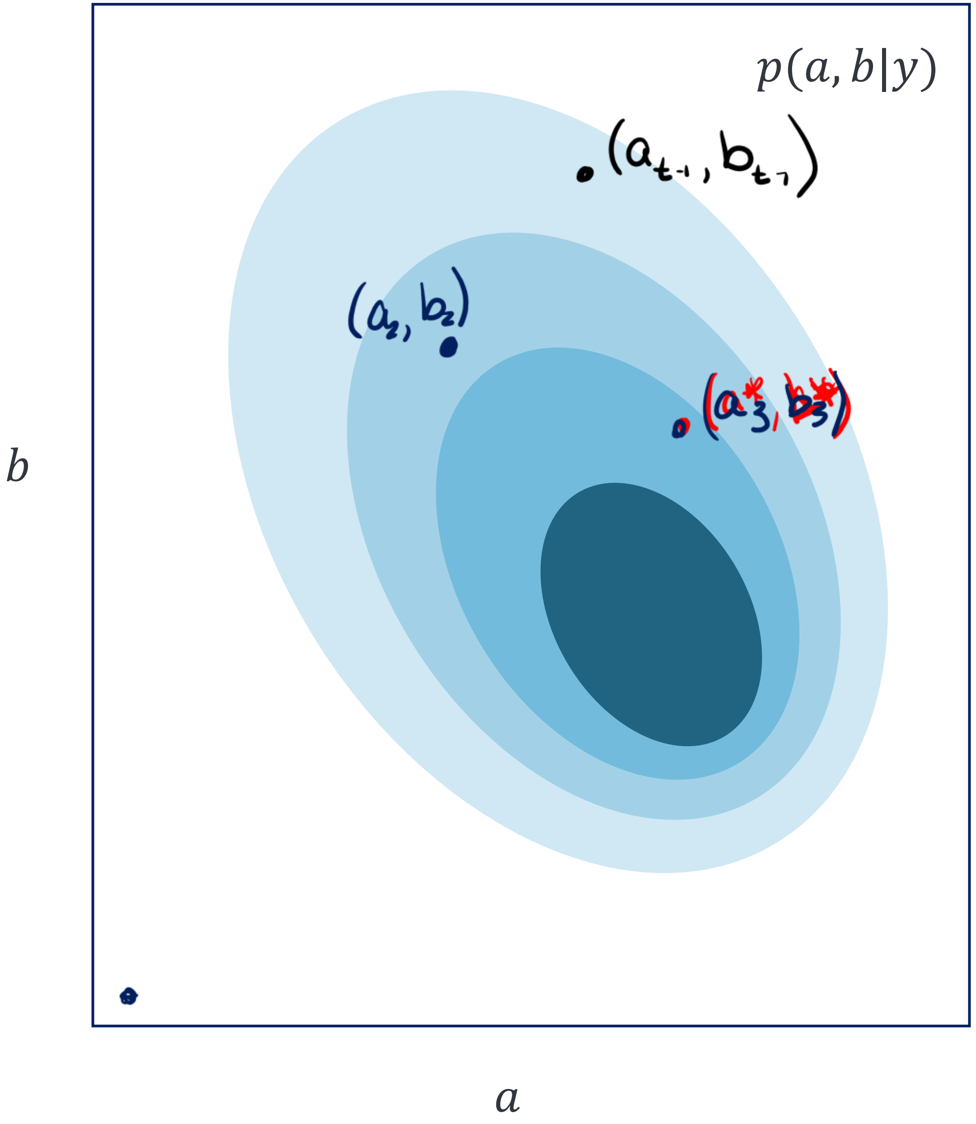
and

Edward Teller,* *Department of Physics, University of Chicago, Chicago, Illinois*

(Received March 6, 1953)

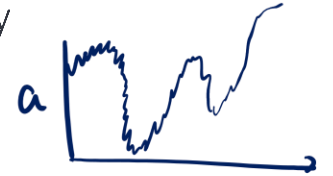
A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional rigid-sphere system have been obtained on the Los Alamos MANIAC and are presented here. These results are compared to the free volume equation of state and to a four-term virial coefficient expansion.

Sketch of how the Metropolis (Rosenbluth) algorithm works for a two-parameter posterior distribution p(a,b|y)



1. Current state is somewhere in parameter space (a_{t-1}, b_{t-1})
2. Suggest a new place in parameter space (a^*, b^*) by making a “jump” according to the proposal distribution
   a. Compare $p(a^*, b^*|y)$ to $p(a_{t-1}, b_{t-1}|y)$
      - Accept a* and b^* if $p(a^*, b^*|y) > p(a_{t-1}, b_{t-1}|y)$
      - If $p(a^*, b^*|y) < p(a_{t-1}, b_{t-1}|y)$ then accept only with some probability
3. If (a^*, b^*) are accepted, then save these in the Markov chain in spot t. If not accepted, then save again (a_{t-1}, b_{t-1}) in spot t
4. Repeat steps 2-3 many times, saving the (a, b) values as you go, to get your Markov chain

a	b
a_{t-1}	b_{t-1}
a_2	b_2
a_3	b_3
a_3	b_3



Markov Chain Monte Carlo (MCMC)

$p(\theta|y) = \frac{p(y|\theta)\,p(\theta)}{p(y)}$

$p(\theta^*|y) = \frac{p(y|\theta^*)\,p(\theta^*)}{p(y)}$

$p(\theta_{old}|y) = \frac{p(y|\theta_{old})\,p(\theta_{old})}{p(y)}$

- Notation
  - θ^* → suggested parameter value(s) (e.g., (a^*, b^*) in our example)
  - θ_{old} → value of parameters saved in previous step of Markov chain

- Accept/reject step
  - If $p(\theta^*|y) > p(\theta_{old}|y)$, then accept θ^* as next value in Markov chain
    - i.e., if $\mathrm{r} = \frac{p(\theta^*|y)}{p(\theta_{old}|y)}$ is greater than 1, then go to that place in parameter space

0.3

  - If $p(\theta^*|y) < p(\theta_{old}|y)$, i.e. $\mathrm{r} = \frac{p(\theta^*|y)}{p(\theta_{old}|y)}$ is less than 1, then …
    - draw a random number x from $\mathrm{Unif}(0,1)$
    - If $x < \frac{p(\theta^*|y)}{p(\theta_{old}|y)}$, then accept θ^*. Otherwise reject θ^* and stay at θ_{old}

Metropolis (Rosenbluth) algorithm → the accept/reject step

- Notation
  - θ^* → suggested parameter value(s) (e.g., (a^*, b^*) in our example)
  - θ_{old} → value of parameters saved in previous step of Markov chain

- Accept/reject step
  - If $p(\theta^*|y) > p(\theta_{old}|y)$, then accept θ^* as next value in Markov chain
    - i.e., if $\mathrm{r} = \frac{p(\theta^*|y)}{p(\theta_{old}|y)}$ $\text{is greater than } 1$, then go to that place in parameter space
  - If $p(\theta^*|y) < p(\theta_{old}|y)$, i.e. $\mathrm{r} = \frac{p(\theta^*|y)}{p(\theta_{old}|y)}$ $\text{is less than } 1$, then …
    - draw a random number x from $\mathrm{Unif}(0,1)$
    - **If $\boldsymbol{x} < \frac{\boldsymbol{p}(\boldsymbol{\theta}^*|\boldsymbol{y})}{\boldsymbol{p}(\boldsymbol{\theta}_{old}|\boldsymbol{y})}$ then accept $\boldsymbol{\theta}^*$**, otherwise reject θ^* and stay at θ_{old}

Metropolis (Rosenbluth) algorithm → the accept/reject step

- $\boxed{\text{If } \boldsymbol{x} < \dfrac{\boldsymbol{p(\theta^*|y)}}{\boldsymbol{p(\theta_{old}|y)}} \text{ then accept } \boldsymbol{\theta^*}}$, otherwise reject θ^* and stay at θ_{old}

- Notice that you don't need the normalization constant for the ratio above:

$$\boldsymbol{r} = \frac{\boldsymbol{p(\theta^*|y)}}{\boldsymbol{p(\theta_{old}|y)}} = \frac{\dfrac{\boldsymbol{p(y|\theta^*)p(\theta^*)}}{\cancel{\boldsymbol{p(y)}}}}{\dfrac{\boldsymbol{p(y|\theta_{old})p(\theta_{old})}}{\cancel{\boldsymbol{p(y)}}}}$$

Metropolis-Hastings Algorithm

Generalizes Metropolis algorithm to an asymmetric jumping (proposal) distribution

- Can be more efficient if you know the target distribution is skewed in some way

Instead of the following used in the accept/reject step

$$r = \frac{p(\theta^*|y)}{p(\theta_{old}|y)}$$

the ratio r is replaced with

$$\boldsymbol{r = \frac{\frac{p(\theta^*|y)}{J_t(\theta^*|\theta_{old})}}{\frac{p(\theta_{old}|y)}{J_t(\theta_{old}|\theta^*)}}}$$

→ This accounts for the asymmetric jumping distribution $\boldsymbol{J_t}$

A helpful visualization tool for different sampling algorithms

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

Posterior Predictive Checking

Posterior predictive distribution

- We can check our model with predictive checks
- The *Posterior predictive distribution* is given by:

$$p\left(y_{future} \middle| y_{observed}\right) = \int p\left(y_{future} \middle| \theta\right) p(\theta | y_{observed}) d\theta$$

- Basic idea:
  - Generate mock data using the posterior distribution of the parameters
  - Compare the mock data to the real data
  - If the model is a good description of the real data, then the model should be able to generate "real" looking data.

Let's go back to our m&m's example to see how this works!

Exercise

- Perform a *posterior predictive check*
  - Draw random θ value from your posterior
  - Given θ, draw random y from the binomial
  - Repeat many times, to get a predictive distribution of y's
  - Predictive distribution of y's → $p(y_{future}|y_{oberved})$
- Plot your predictive distribution of y's, compare to your neighbour's