Lecture 9
Metropolis Sampler and Markov Chains
MCMC: Markov Chain Monte Carlo

Last time: Simulated Annealing

Minimize f by identifying with the energy of an imaginary physical system undergoing an annealing process.

Move from x_i to x_j via a **proposal**.

If the new state has lower energy, accept x_j.

If the new state has higher energy, accept with probability

$$A = \exp\left(-\Delta f / kT\right)$$

Today

- from annealing to Metropolis
- markov chains and MCMC
- Metropolis and an introduction to Metropolis-Hastings

Annealing Recap

- stochastic acceptance of higher energy states, allows our process to escape local minima.
- high T local minima discouraged
- low T only few uphill moves
- Thus, if we get our temperature decrease schedule right, we can hope that we will converge to a global minimum.

If the lowering of the temperature is sufficiently slow, the system reaches "thermal equilibrium" at each temperature. Then Boltzmann's distribution applies:

$$p(X = i) = \frac{1}{Z(T)} \exp\left(\frac{-E_i}{kT}\right)$$

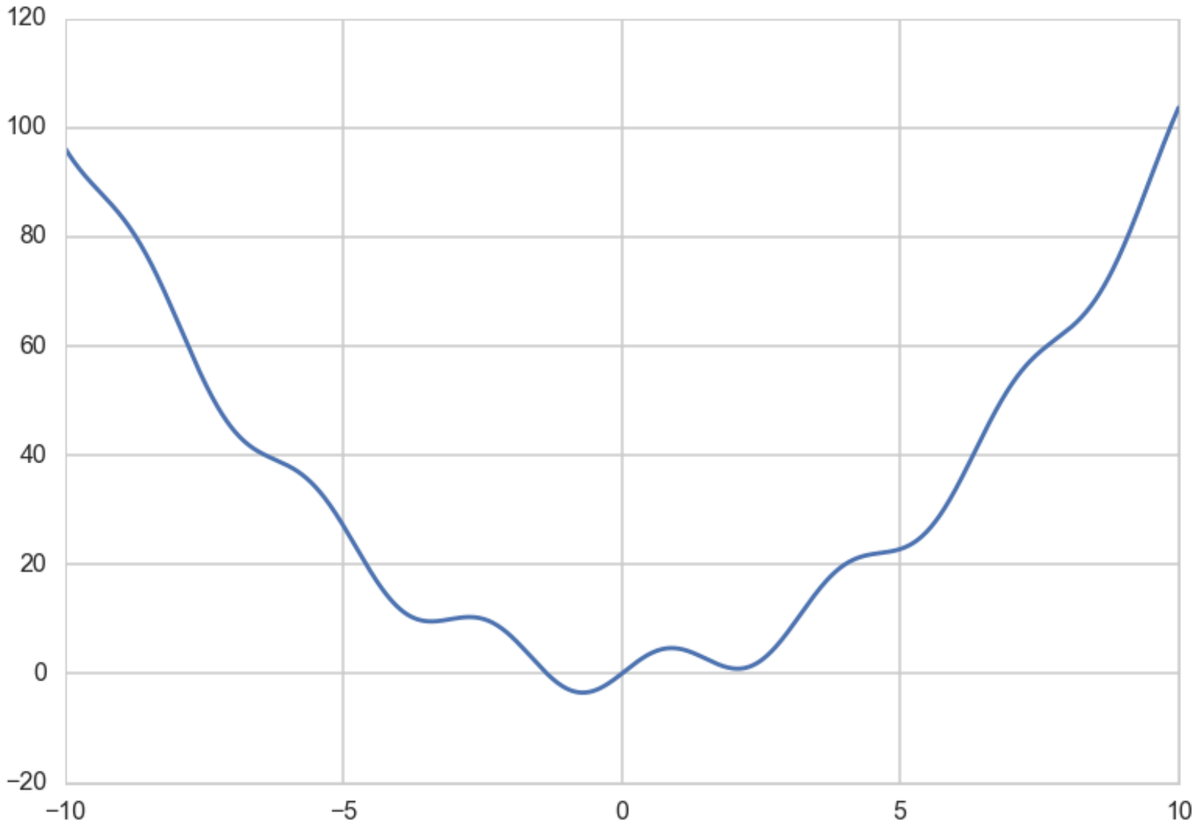
where

$$Z(T) = \sum_j \exp\left(\frac{-E_j}{kT}\right)$$

Proposal

- it proposes a new position x from a **neighborhood** $\mathcal{N}$ at which to evaluate the function.
- all the positions x in the domain we wish to minimize a function f over ought to be able to communicate.
- detailed balance: proposal is symmetric
- ensures $\{x_t\}$ generated by simulated annealing is a stationary markov chain with target boltzmann distribution: equilibrium

Example: $x^2 + 4sin(2x)$

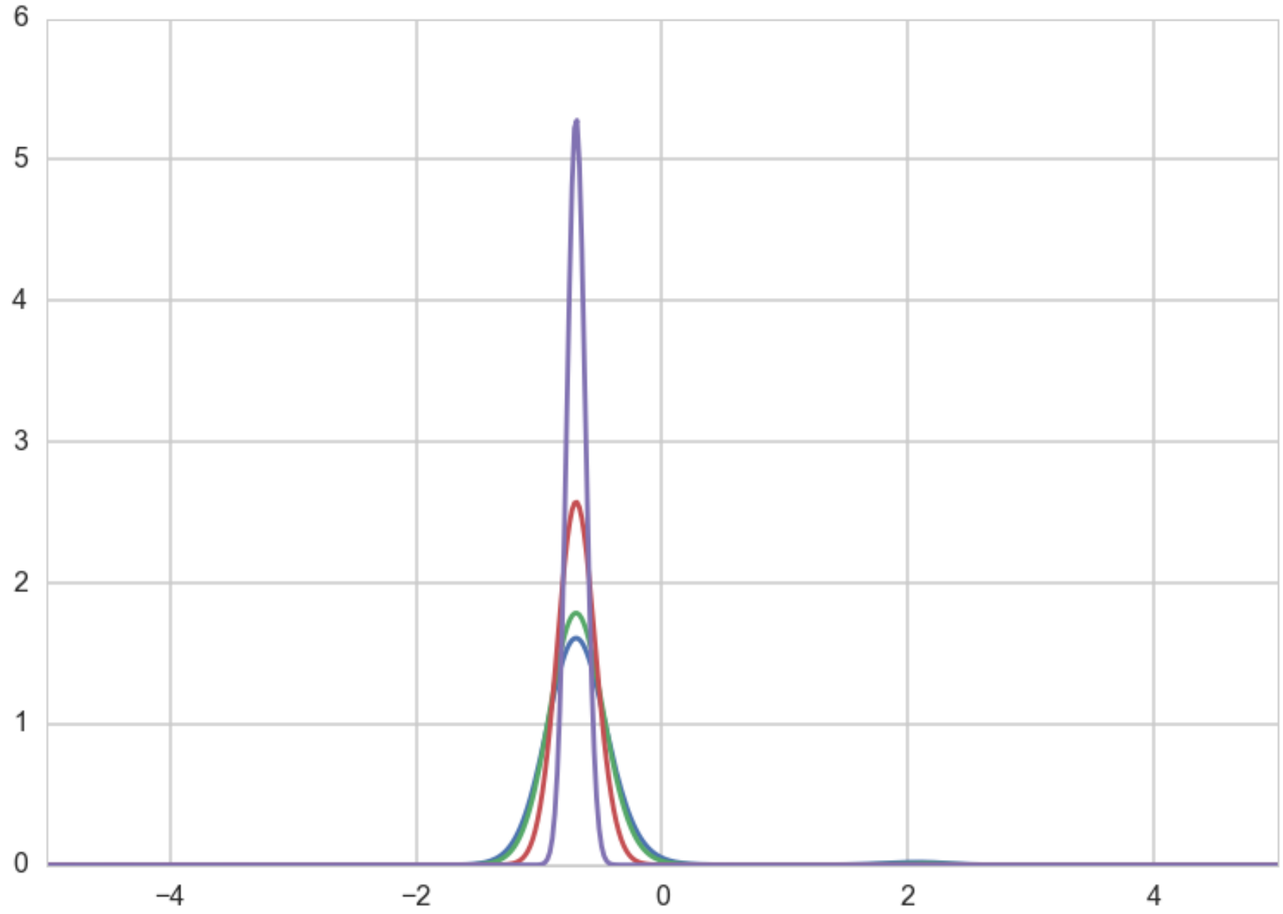


If you identify

$$p_T(x) = e^{-f(x)/T} \text{ and } p(x) = e^{-f(x)}$$

Then:

$$P_T(x) = P(x)^{1/T}$$



Normalized Boltzmann distribution

- M global minima in set $\mathcal{M}$
- function minimum value f_{min}:

$$p(x_i) = \frac{e^{-(f(x_i)-f_{min})/T}}{M + \sum_{j \notin \mathcal{M}} e^{-(f(x_i)-f_{min})/T}}$$

As $T \to 0$ from above, this becomes $1/M$ if $x_i \in \mathcal{M}$ and 0 otherwise.

Sampling a Distribution

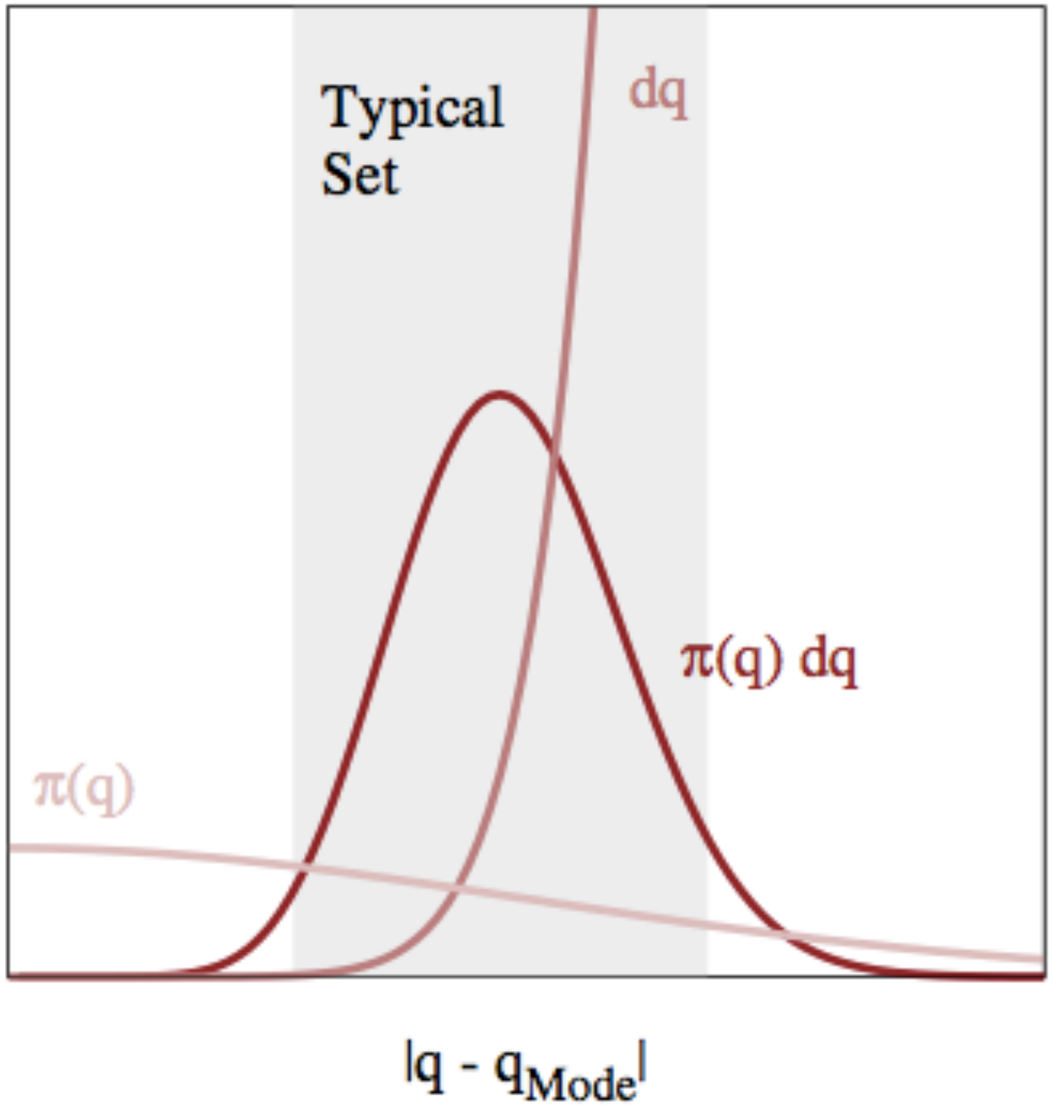
- Turn the question on its head.
- Suppose we wanted to sample from a distribution $p(x)$ (corresponding to a minimization of energy $-log(p(x))$).
- keep our symmetric proposal (reversibility!). Need irreducibility to sample from full distribution
- set T=1, and use our simulated annealing method

```
def metropolis(p, qdraw, nsamp, xinit):
    samples=np.empty(nsamp)
    x_prev = xinit
    for i in range(nsamp):
        x_star = qdraw(x_prev)
        p_star = p(x_star)
        p_prev = p(x_prev)
        pdfratio = p_star/p_prev
        if np.random.uniform() < min(1, pdfratio):
            samples[i] = x_star
            x_prev = x_star
        else:#we always get a sample
            samples[i]= x_prev

    return samples
```

Uniform Proposal to sample the standard gaussian

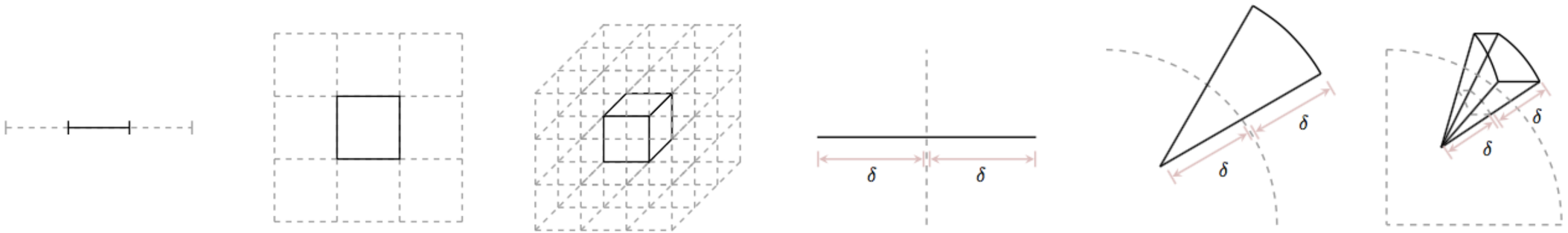
```
from scipy.stats import uniform
def propmaker(delta):
    rv = uniform(-delta, 2*delta)
    return rv
uni = propmaker(0.5)
def uniprop(xprev):
    return xprev+uni.rvs()
```



Why do this?

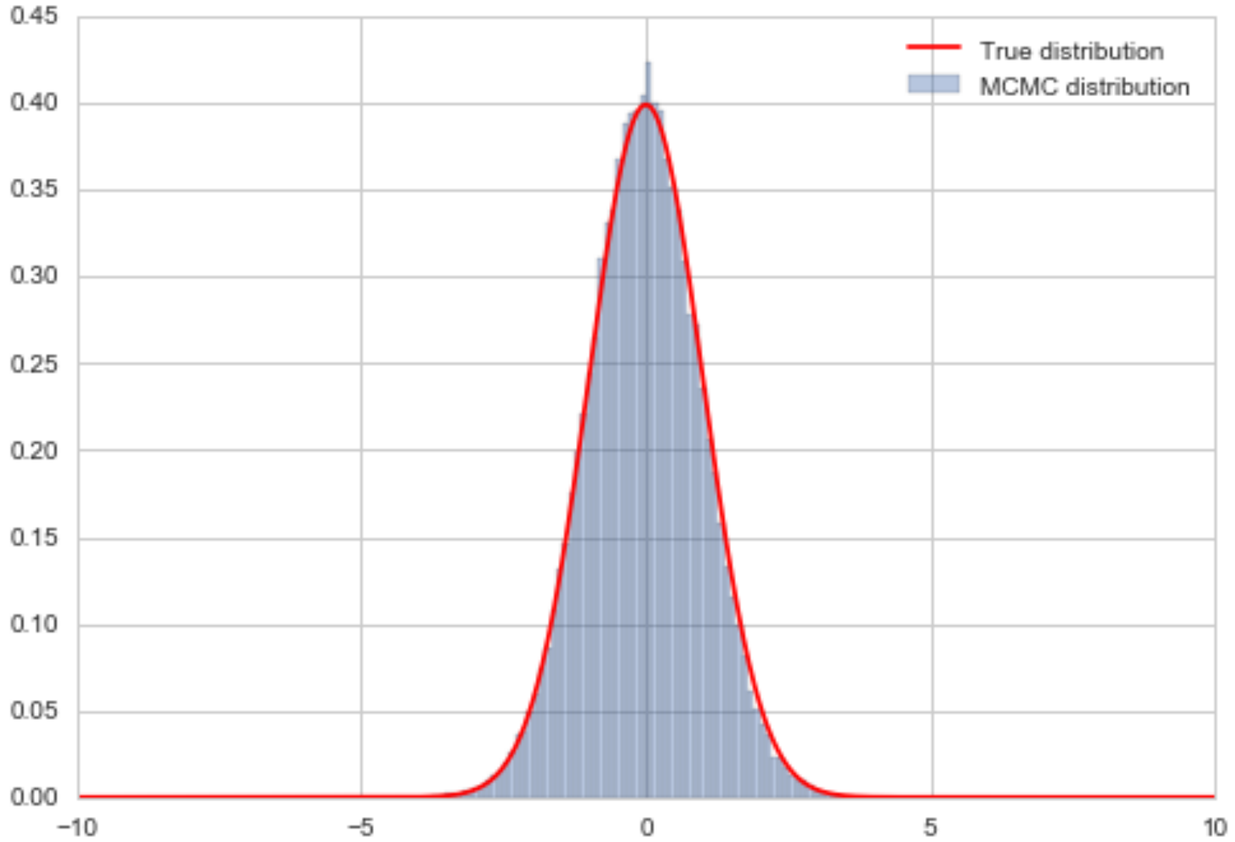
- Why not rejection? wasteful
- more wasteful in higher dimensions
- curse of dimensionality in higher dimensions
- volume around mode gets smaller
- interplay of density and volume

Curse of dimensionality



as dimensionality increases, center is lower volume, outside has more volume

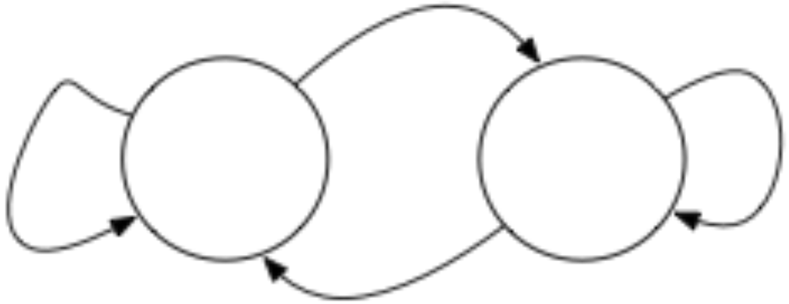
Sampling from gaussian with uniform proposal



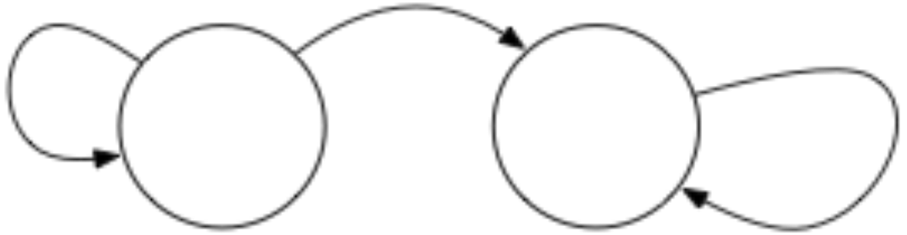
Markov Chain

$$T(x_n|x_{n-1}, x_{n-1}\ldots, x_1) = T(x_n|x_{n-1})$$

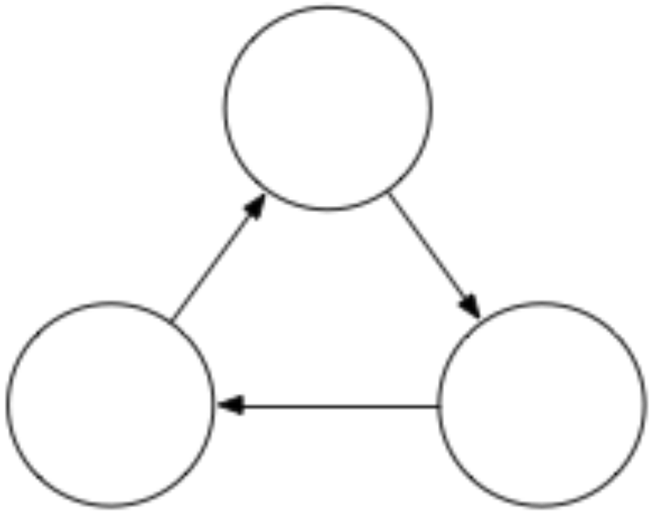
- non IID, stochastic process
- but one step memory only
- widely applicable, first order equations



Irreducible



Reducible



Irreducible, but period 3

Stationarity

$$sT = s \text{ or } \sum_i s_i T_{ij} = s_j \text{ or}$$

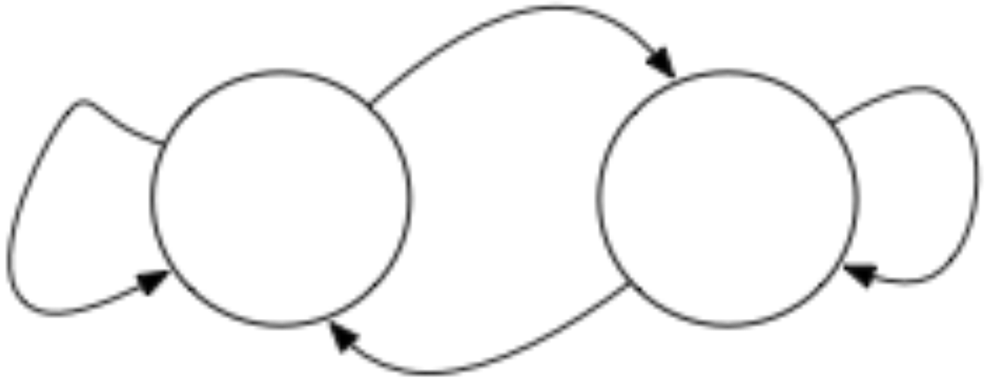
Continuous case: define T so that:

$$\int dx_i \, s(x_i) T(x_{i+1}|x_i) = s(x_{i+1}) \text{ then}$$

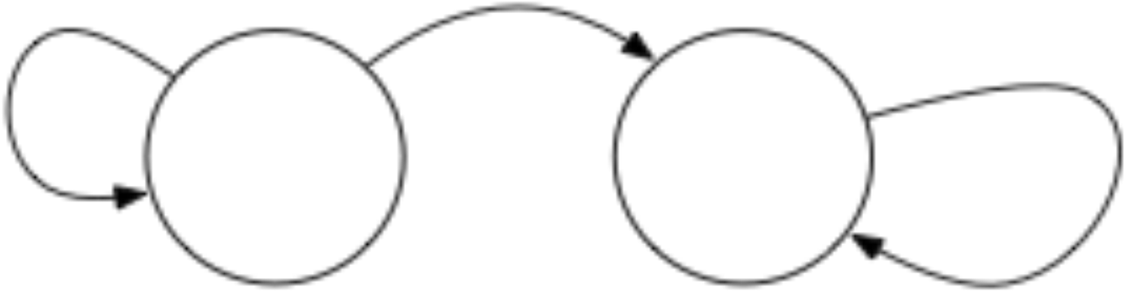
$$\int dx s(x) T(y|x) = \int p(y,x) dx = s(y)$$

Jargon

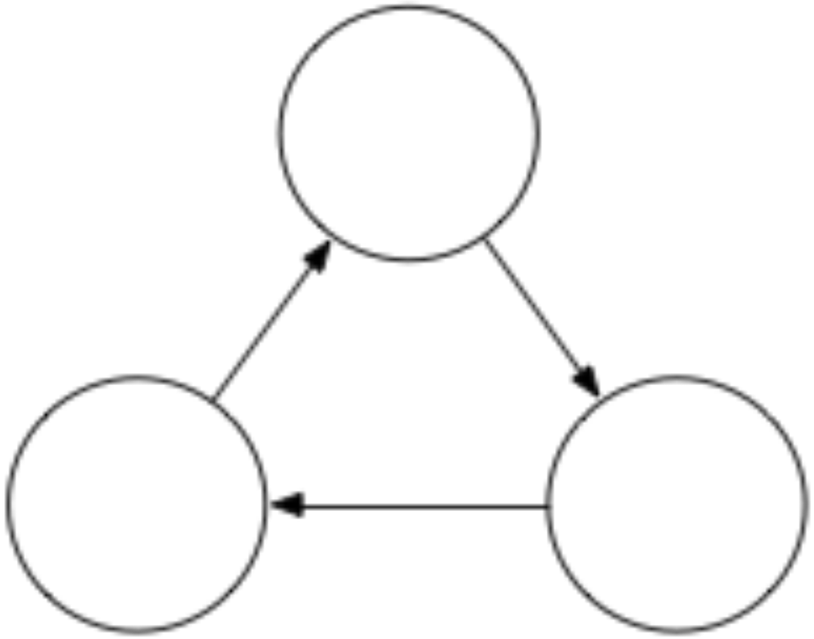
- **Irreducible**: can go from anywhere to everywhere
- **Aperiodic**: no finite loops
- **Recurrent**: visited repeatedly. Harris recurrent if all states are visited infinitely as $t \to \infty$.



Irreducible



Reducible



Irreducible, but period 3

Stationarity, again

A irreducible (goes everywhere) and aperiodic (no cycles) markov chain will eventually converge to a stationary markov chain. It is the marginal distribution of this chain that we want to sample from, and which we do in metropolis (and for that matter, in simulated annealing).

$$\int dx s(x) T(y|x) = \int p(y,x) dx = s(y)$$

BURNIN

Ergodicity (stronger statement)

$$\int g(x)f(x)dx = \frac{1}{N}\sum_{j=B+1}^{B+N} g(x_j)$$

Aperiodic, irreducible, positive Harris recurrent markov chains are ergodic, that is, in the limit of infinite (many) steps, the marginal distribution of the chain is the same.

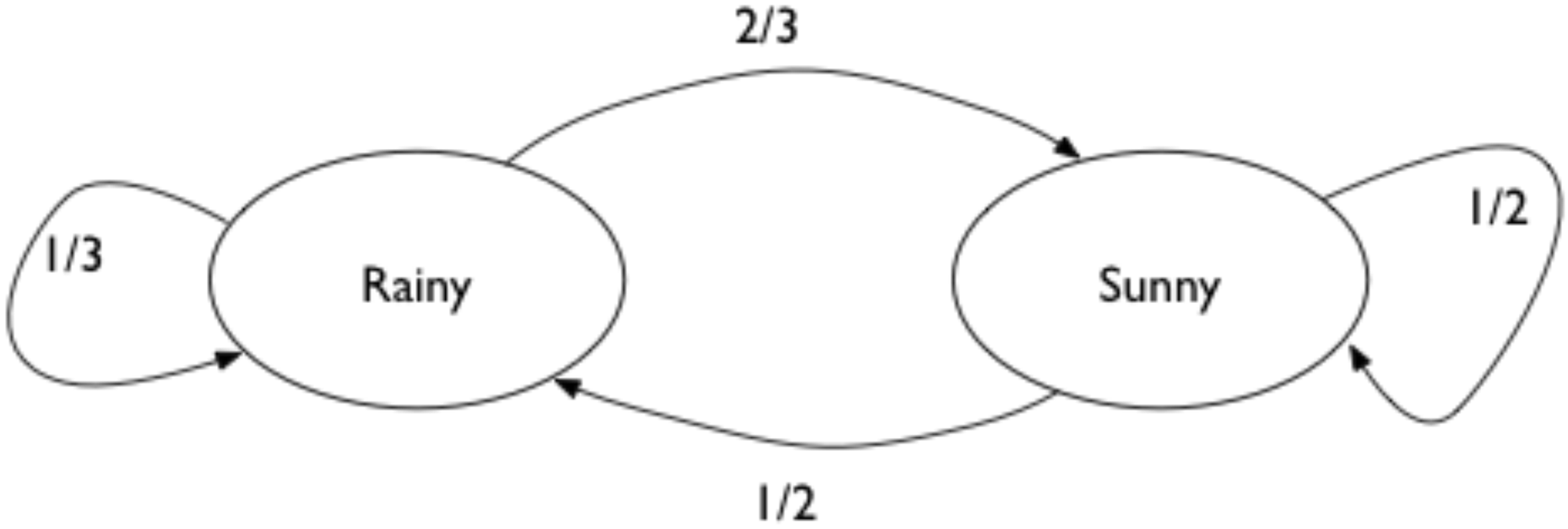
Detailed balance is enough for stationarity

$$s(x)T(y|x) = s(y)T(x|y)$$

If one sums both sides over x

$\int dx s(x)t(y|x) = s(y) \int dx T(x|y)$ which gives us back the stationarity condition from above.

Rainy Sunny Markov chain



aperiodic and irreducible

Transition matrix, applied again and again

```
array([[ 0.33333333,  0.66666667],
       [ 0.5       ,  0.5       ]])


[[ 0.44444444  0.55555556]
 [ 0.41666667  0.58333333]]
-----------------
[[ 0.42592593  0.57407407]
 [ 0.43055556  0.56944444]]
-----------------
[[ 0.42901235  0.57098765]
 [ 0.42824074  0.57175926]]
-----------------
[[ 0.42849794  0.57150206]
 [ 0.42862654  0.57137346]]
-----------------
[[ 0.42858368  0.57141632]
 [ 0.42856224  0.57143776]]
```

Stationary distribution can be solved for:

Assume that it is $s = [p, 1-p]$

Then: $sT = s$

gives us

$$p \times (1/3) + (1-p) \times 1/2 = p$$

and thus $p = 3/7$

```
np.dot([0.9,0.1], tm_before): array([ 0.42858153,  0.57141847])
```

MCMC

- Markov Chain Monte Carlo
- Footing for Metropolis

Find a markov chain whose stationary distribution is the distribution we need to sample from

As long detailed balance we are ok:

$$s(x_i)T(x_{i-1}|x_i) = s(x_{i-1})T(x_i|x_{i-1})$$

Transition matrix for Metropolis:

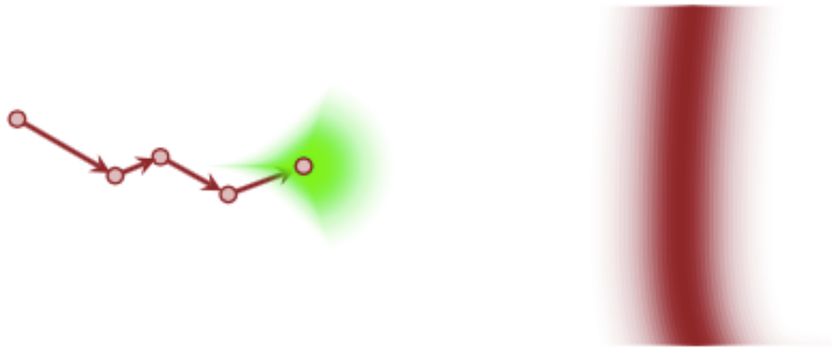
$$T(x_i|x_{i-1}) = q(x_i|x_{i-1})\, A(x_i, x_{i-1}) + \delta(x_{i-1} - x_i) r(x_i)$$ where

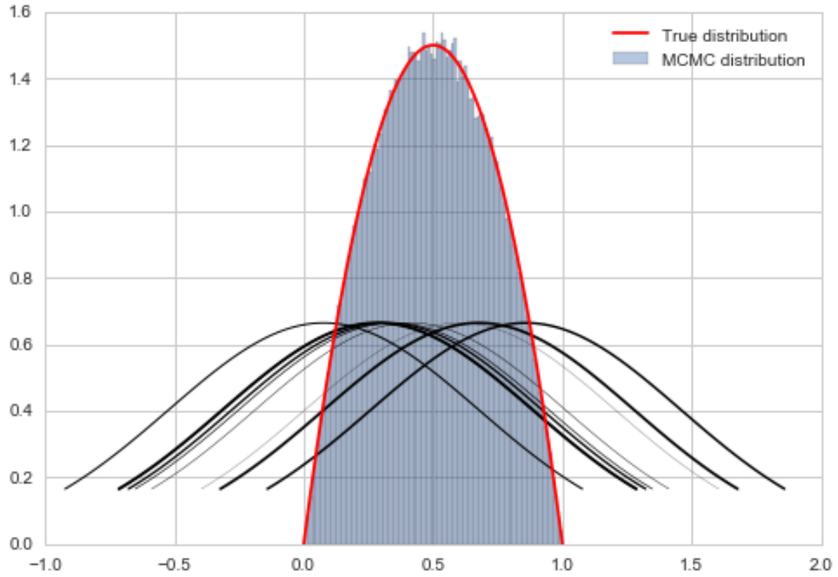
$$A(x_i, x_{i-1}) = min(1, \frac{s(x_i)}{s(x_{i-1})})$$

is the Metropolis acceptance probability and

$r(x_i) = \int dy q(y|x_i)(1 - A(y, x_i))$ is the rejection term.

Intuition: approaches typical set



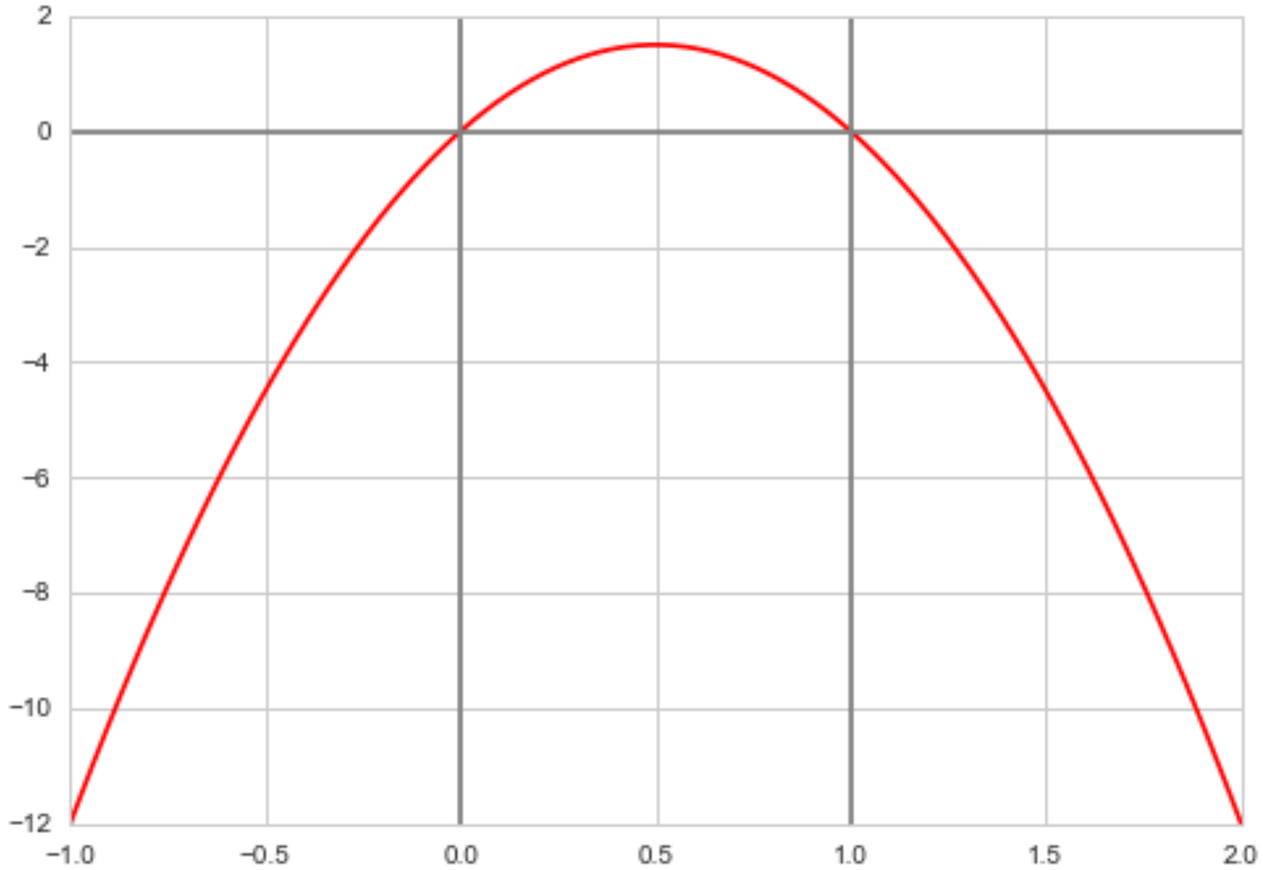


Instead of sampling p we sample q, yielding a new state, and a new proposal distribution from which to sample.

- The possibility of rejection in the Metropolis algorithm based on the throw of a random uniform makes the chain aperiodic.
- And if we want it to be irreducible, we need to make sure q can go everywhere that p can, or that the support of q includes everywhere the support of p

Thus our Metropolis algorithm converges.

Metropolis-Hastings



- notice tails
- works on metropolis because we compare uniform to negative
- we could reject but this is wrong
- leads to asymmetric proposal
- might want to use a positive, 0-1 distribution like beta anyway. But asymmetric.

Metropolis-Hastings

```
def metropolis_hastings(p,q, qdraw, nsamp, xinit):
    samples=np.empty(nsamp)
    x_prev = xinit
    for i in range(nsamp):
        x_star = qdraw(x_prev)
        p_star = p(x_star)
        p_prev = p(x_prev)
        pdfratio = p_star/p_prev
        proposalratio = q(x_prev, x_star)/q(x_star, x_prev)
        if np.random.uniform() < min(1, pdfratio*proposalratio):
            samples[i] = x_star
            x_prev = x_star
        else:#we always get a sample
            samples[i]= x_prev

    return samples
```