data analysis with

Markov chain Monte Carlo

Dan Foreman-Mackey
CCPP@NYU

Dan Foreman-Mackey

dan.iel.fm

dfm

@exoplaneteer

I'm a **grad student**
in **Camp Hogg**
at **NYU**

David W. Hogg

I'm an **engineer**
in **Camp Hogg**
at **NYU**

David W. Hogg

I build **tools**
for **data analysis**
in **astronomy** (mostly)

David W. Hogg

introducing **emcee** the MCMC Hammer

arxiv.org/abs/1202.3665

dan.iel.fm/emcee

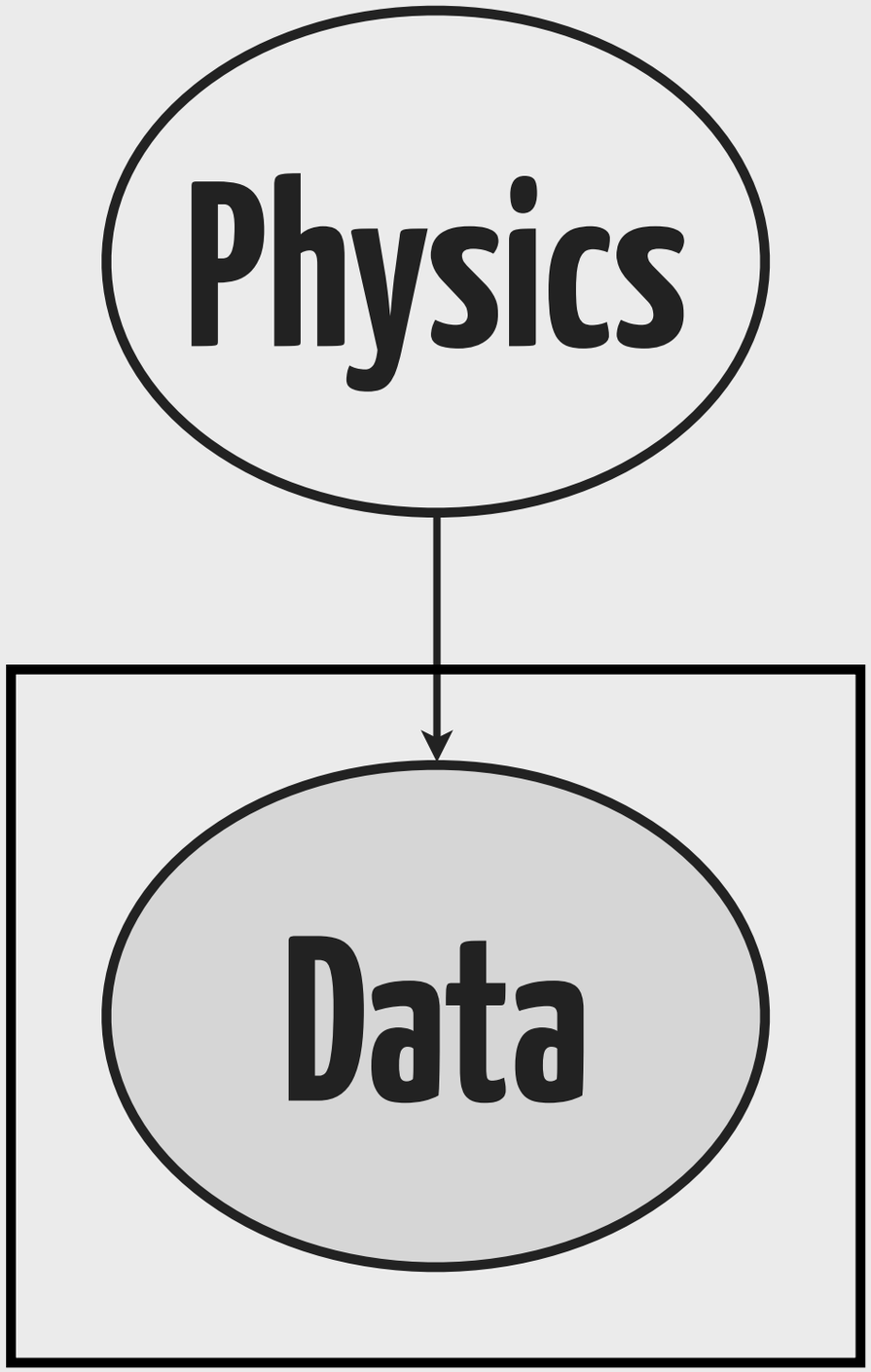
I work on

the Local Group
exoplanets
variable stars
image modeling
calibration

I work on

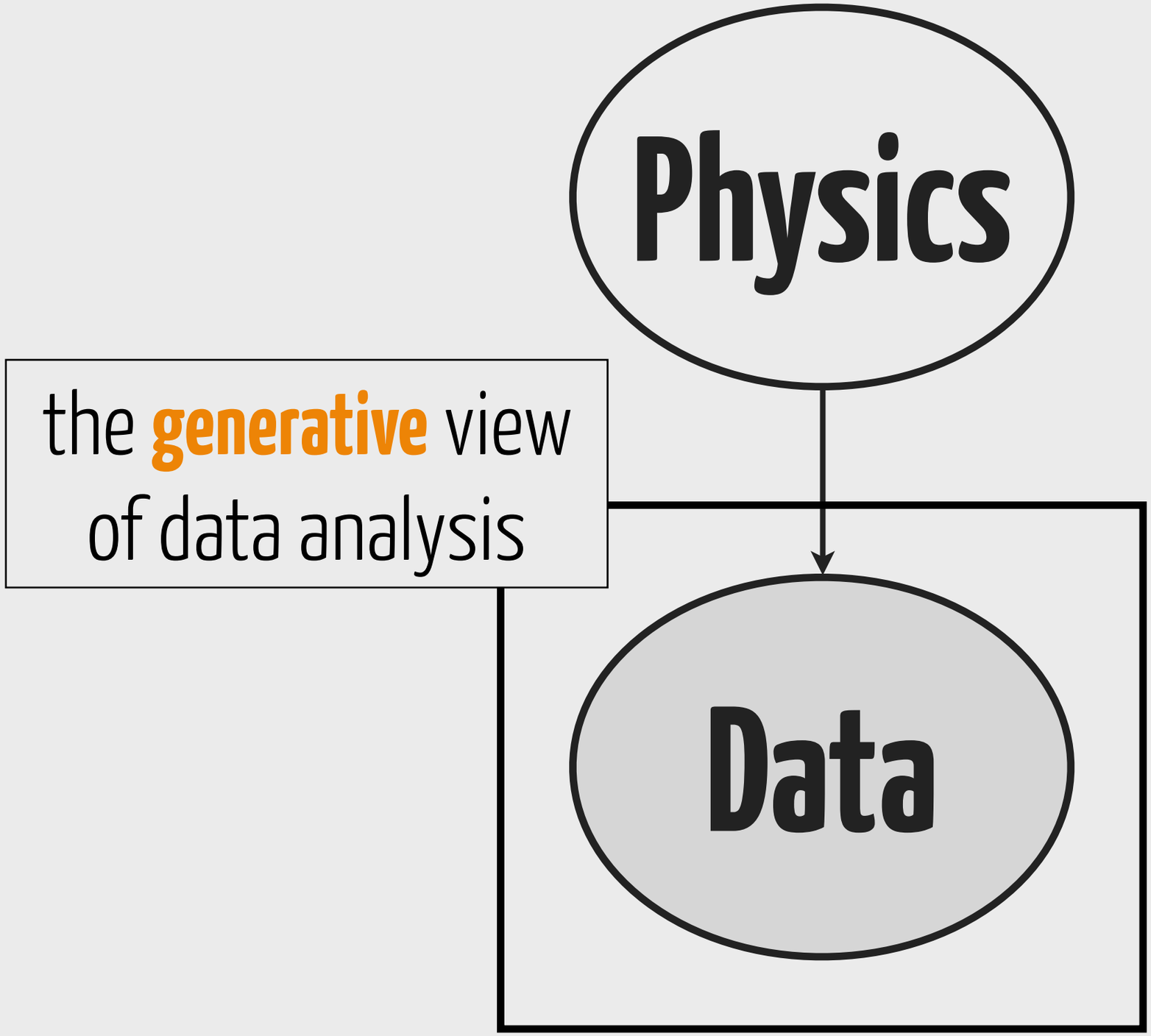
the Local Group
exoplanets
variable stars
image modeling
calibration
not writing papers

I work on **the Local Group**
exoplanets
variable stars
image modeling
calibration
not writing papers



The graphical model of my research.

a sketch of



The graphical model of my research.

a sketch of

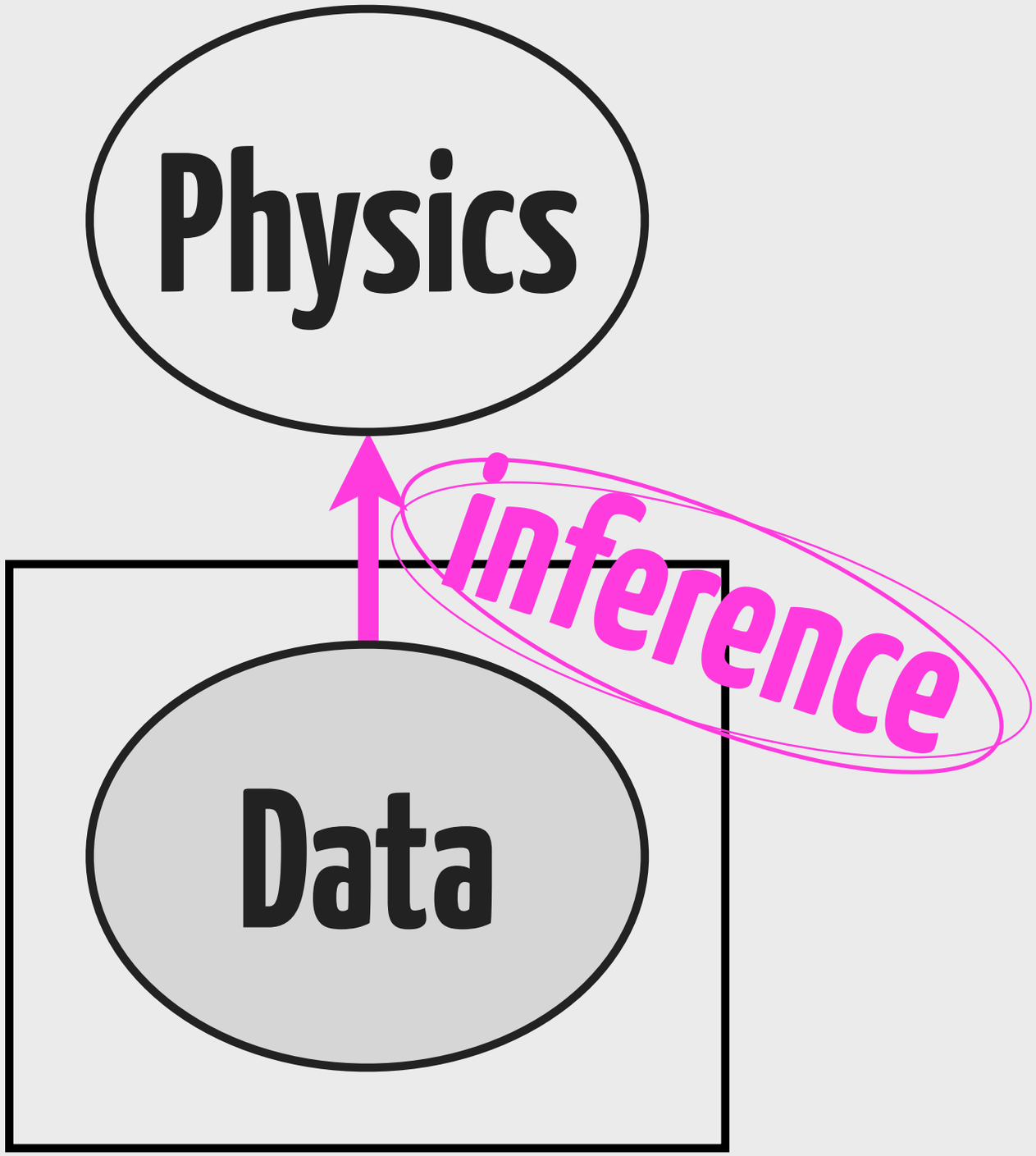
for example

$$\chi^2 = \sum_{n=1}^{N} \frac{[y_n - \overbrace{(m\,x_n + b)}^{\text{a line}}]^2}{\sigma_n^2}$$

$$p(\{y_n,\, x_n,\, \sigma_n^2\}\,|\,m,\, b) \propto \exp\left(-\frac{1}{2}\,\chi^2\right)$$

synthetic data: $\hat{y}_n = m\,x_n + b + \epsilon_n$

noise model: $\epsilon_n \sim \mathcal{N}(0;\, \sigma_n^2)$



The graphical model of my research.

a sketch of

$$p(\mathrm{data} \mid \mathrm{physics})$$

likelihood function/generative model

$$p(\mathrm{physics} \mid \mathrm{data}) \propto p(\mathrm{physics})\, p(\mathrm{data} \mid \mathrm{physics})$$

posterior probability

good/bad news:

but **physics**

is **everything**.

including **things we don't care about!**

a more realistic model:

cool physics

$$p(\mathbf{D} \mid \boldsymbol{\theta}, \boldsymbol{\alpha})$$

data

garbage

what if we underestimated our error bars?

(by some additive variance)

"physics" "jitter" (not physics)

$$p(\{y_n,\, x_n,\, \sigma_n^2\}\,|\,\overline{m,\, b,}\, \delta^2) =$$

$$\prod_{n=1}^{N} \frac{1}{\sqrt{2\,\pi\,(\sigma_n^2 + \delta^2)}} \exp\left[-\frac{1}{2}\frac{[y_n - (m\,x_n + b)]^2}{\sigma_n^2 + \delta^2}\right]$$

Do The Right Thing™

$$p(\mathbf{D} \mid \boldsymbol{\theta}) \propto \int p(\boldsymbol{\alpha})\, p(\mathbf{D} \mid \boldsymbol{\theta}, \boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\alpha}$$

marginalize.

in our example

$$p(\{y_n,\, x_n,\, \sigma_n\}\,|\,m,\, b) =$$

$$\int \mathrm{d}\delta^2\, p(\delta^2) \prod_{n=1}^{N} \frac{1}{\sqrt{2\,\pi\,(\sigma_n^2 + \delta^2)}} \exp\left[-\frac{1}{2}\,\frac{[y_n - (m\,x_n + b)]^2}{\sigma_n^2 + \delta^2}\right]$$

BOOM!

What do you **really** want?

What do you **really** want?

$$E_p[f(\boldsymbol{\theta})] = \frac{1}{Z} \int f(\boldsymbol{\theta})\, p(\boldsymbol{\theta})\, p(\mathbf{D} \mid \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}$$

What do you **really** want?

$$E_p[f(\boldsymbol{\theta})] = \frac{1}{Z} \int f(\boldsymbol{\theta})\, p(\boldsymbol{\theta})\, p(\mathbf{D} \mid \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}$$

probably.

marginalization

$$p(\mathbf{D} \mid \boldsymbol{\theta}) \propto \int p(\boldsymbol{\alpha})\, p(\mathbf{D} \mid \boldsymbol{\theta}, \boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\alpha}$$

expectation

$$E_p[f(\boldsymbol{\theta})] = \frac{1}{Z} \int f(\boldsymbol{\theta})\, p(\boldsymbol{\theta})\, p(\mathbf{D} \mid \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}$$

marginalization

$$p(\mathbf{D} \mid \boldsymbol{\theta}) \propto \int p(\boldsymbol{\alpha})\, p(\mathbf{D} \mid \boldsymbol{\theta}, \boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\alpha}$$

large number of dimensions

expectation

$$E_p[f(\boldsymbol{\theta})] = \frac{1}{Z} \int f(\boldsymbol{\theta})\, p(\boldsymbol{\theta})\, p(\mathbf{D} \mid \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}$$

marginalization

$$p(\mathbf{D} \mid \boldsymbol{\theta}) \propto \int p(\boldsymbol{\alpha})\, p(\mathbf{D} \mid \boldsymbol{\theta}, \boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\alpha}$$

large number of dimensions

expectation

whoa!

$$E_p[f(\boldsymbol{\theta})] = \frac{1}{Z} \int f(\boldsymbol{\theta})\, p(\boldsymbol{\theta})\, p(\mathbf{D} \mid \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}$$

marginalization

$$p(\mathbf{D} \mid \boldsymbol{\theta}) \propto \int p(\boldsymbol{\alpha})\, p(\mathbf{D} \mid \boldsymbol{\theta}, \boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\alpha}$$

large number of dimensions

expectation

whoa!

$$E_p[f(\boldsymbol{\theta})] = \frac{1}{Z} \int f(\boldsymbol{\theta})\, p(\boldsymbol{\theta})\, p(\mathbf{D} \mid \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}$$

This is HARD.

(in general)

OK

now that we agree...

as you learned in middle school

$$\int f(\mathbf{x})\, p(\mathbf{x})\, \mathrm{d}\mathbf{x} \approx \frac{1}{N} \sum_{n=1}^{N} f(\mathbf{x}_n)$$

where: $\mathbf{x}_n \sim p(\mathbf{x})$

error: $\delta \propto \frac{1}{\sqrt{N'}}$

number of **independent** samples

as you learned in middle school

$$\int f(\mathbf{x})\, p(\mathbf{x})\, \mathrm{d}\mathbf{x} \approx \frac{1}{N} \sum_{n=1}^{N} f(\mathbf{x}_n)$$

where: $\mathbf{x}_n \sim p(\mathbf{x})$

error: $\delta \propto \frac{1}{\sqrt{N'}}$

number of **independent** samples

MCMC

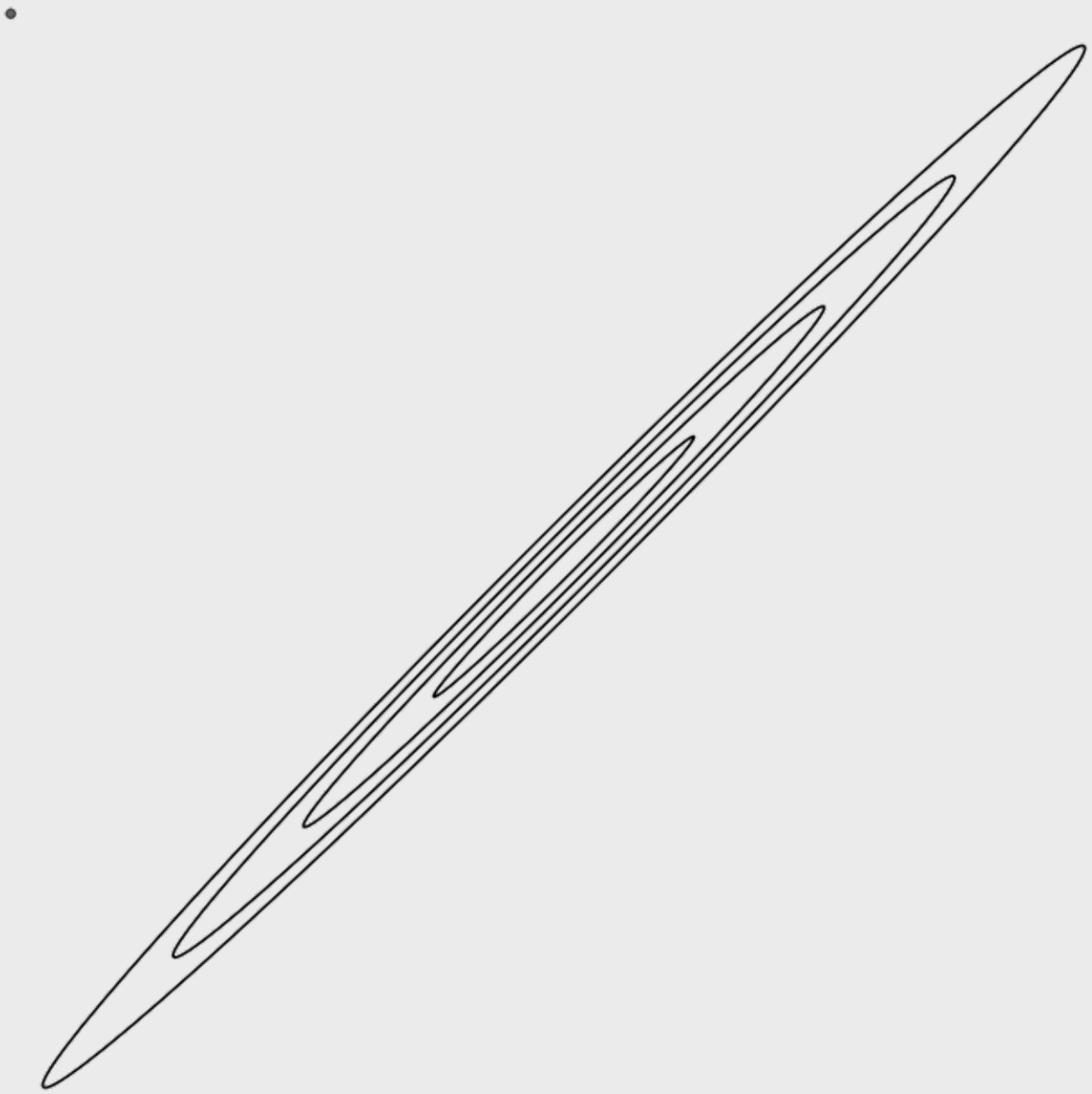
draws samples from a probability function

and all you need to be able to do is

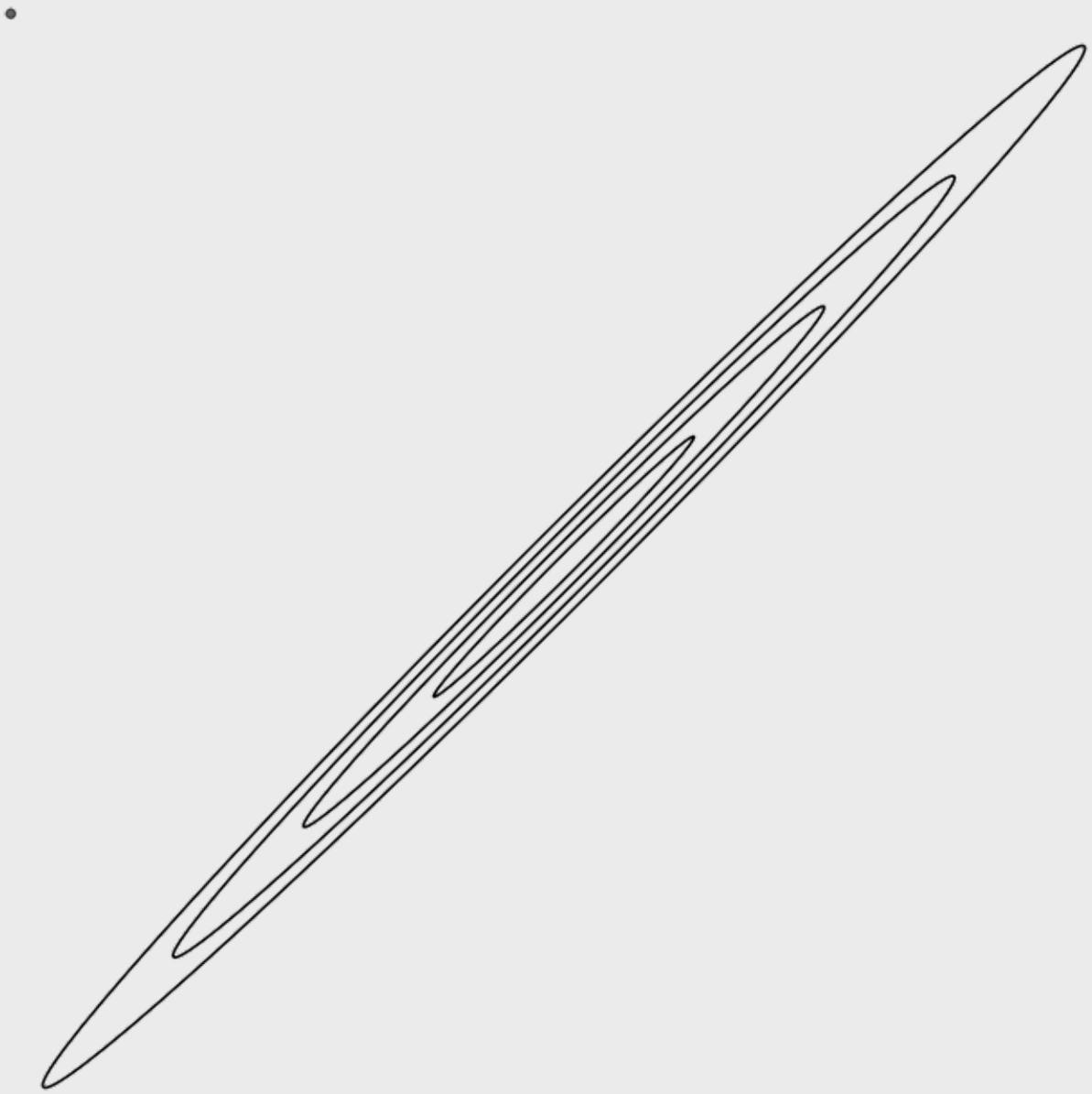
evaluate

the function

(up to a constant)



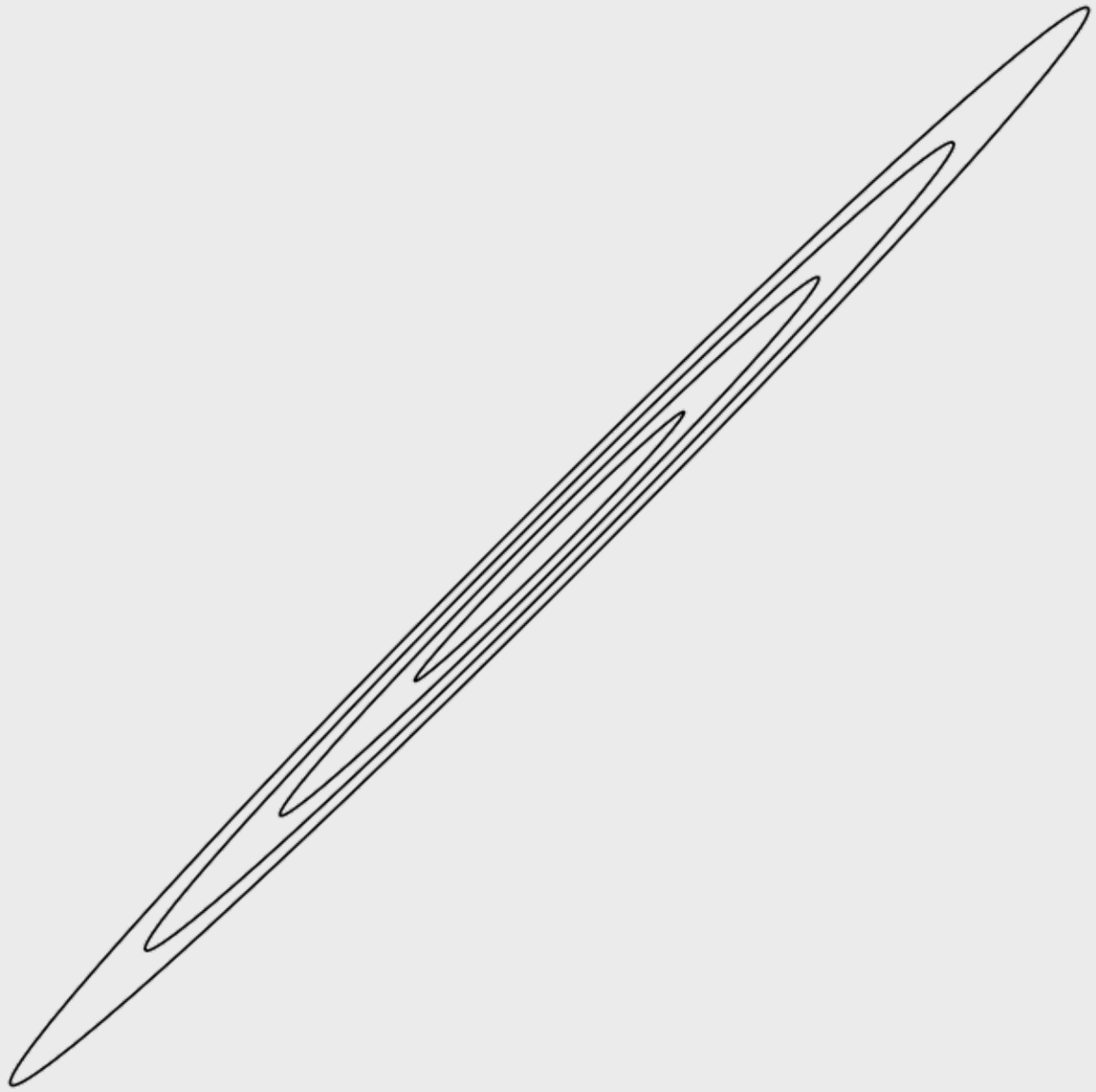
Metropolis–Hastings



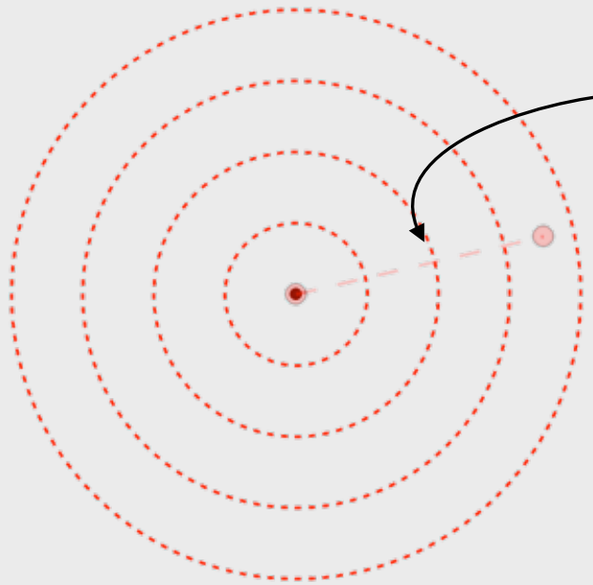
Metropolis–Hastings

in an ideal world

start here
perhaps

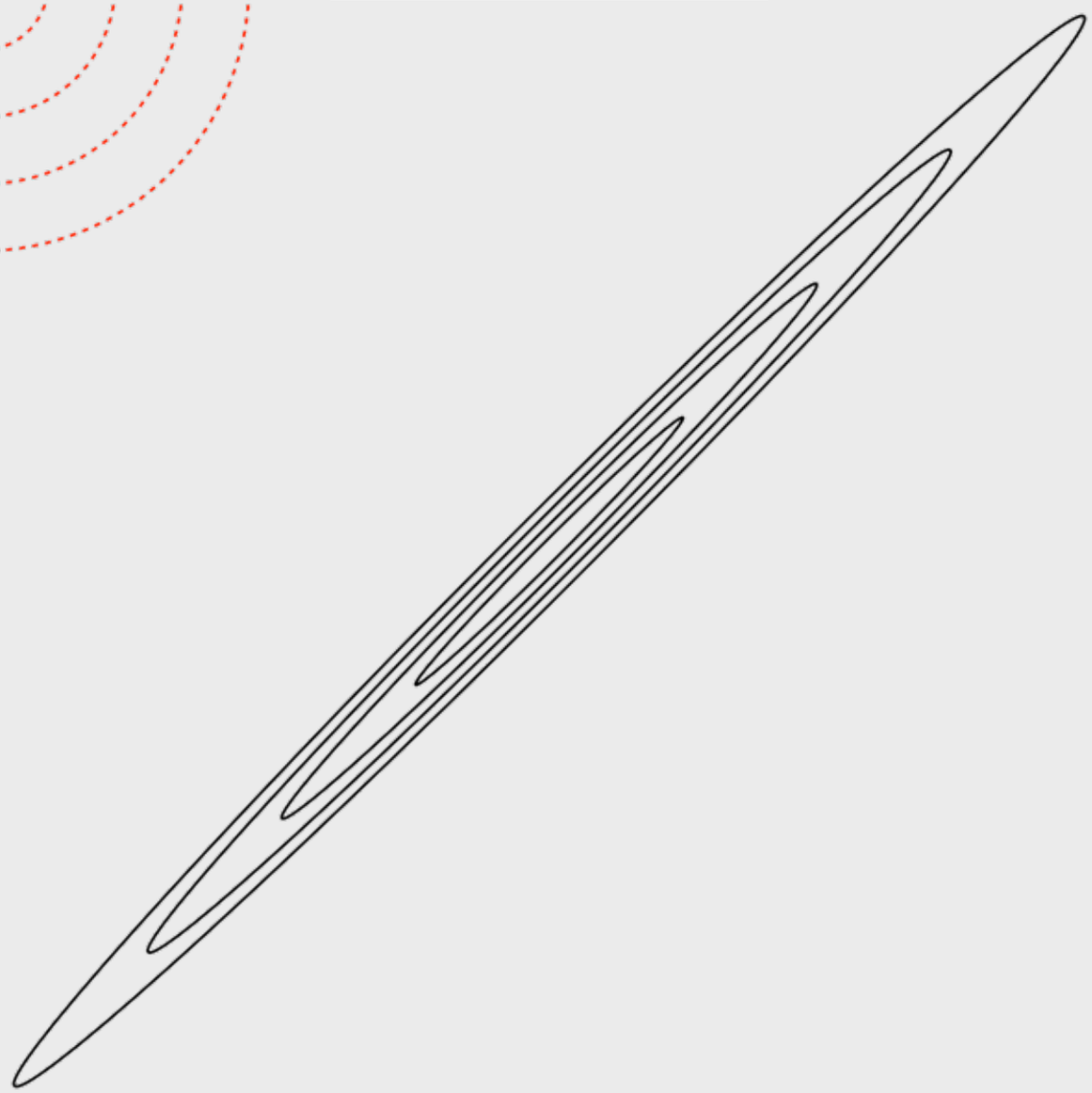


Metropolis–Hastings
in an ideal world

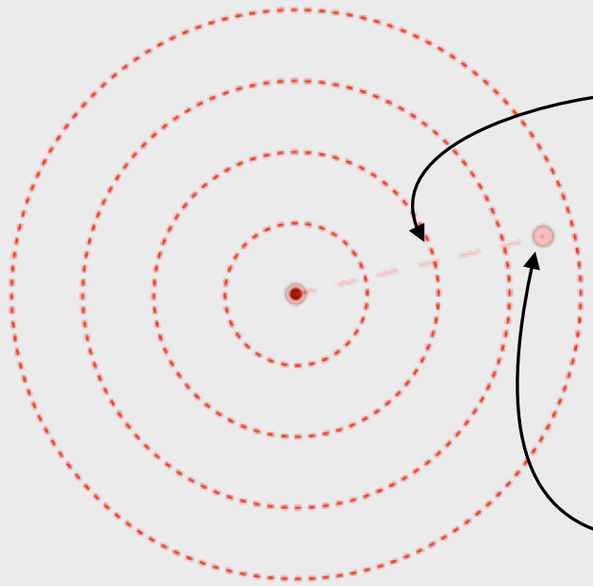


propose
a new position

$$\mathbf{x}' \sim q(\mathbf{x}'; \mathbf{x})$$



Metropolis–Hastings
in an ideal world



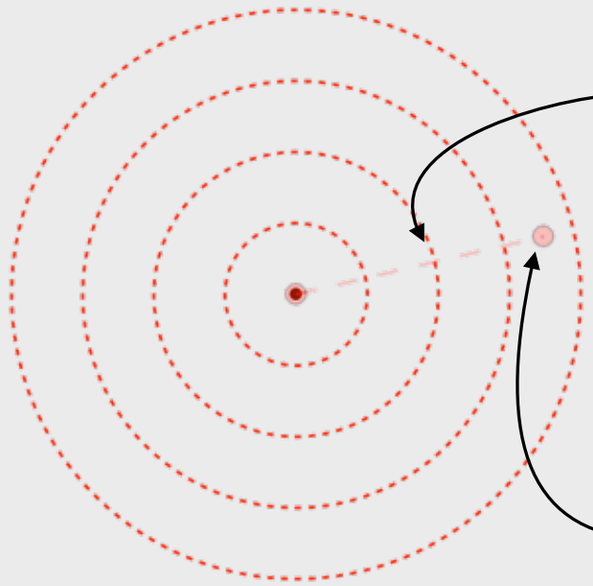
propose
a new position

$$\mathbf{x}' \sim q(\mathbf{x}'; \mathbf{x})$$

accept?

$$p(\text{accept}) = \min\left(1, \frac{p(\mathbf{x})}{p(\mathbf{x}')} \frac{q(\mathbf{x}; \mathbf{x}')}{q(\mathbf{x}'; \mathbf{x})}\right)$$

Metropolis–Hastings
in an ideal world



propose
a new position

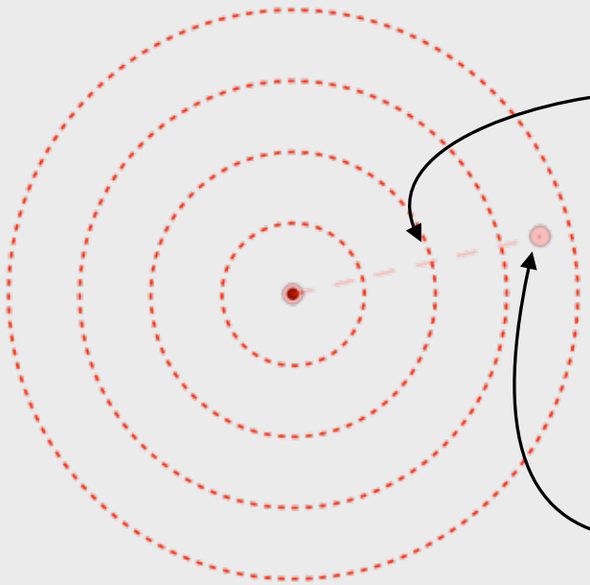
$$\mathbf{x}' \sim q(\mathbf{x}'; \mathbf{x})$$

accept?

$$p(\text{accept}) = \min\left(1, \frac{p(\mathbf{x})}{p(\mathbf{x}')} \frac{q(\mathbf{x}; \mathbf{x}')}{q(\mathbf{x}'; \mathbf{x})}\right)$$

definitely.

Metropolis–Hastings
in an ideal world



propose
a new position

$$\mathbf{x}' \sim q(\mathbf{x}'; \mathbf{x})$$

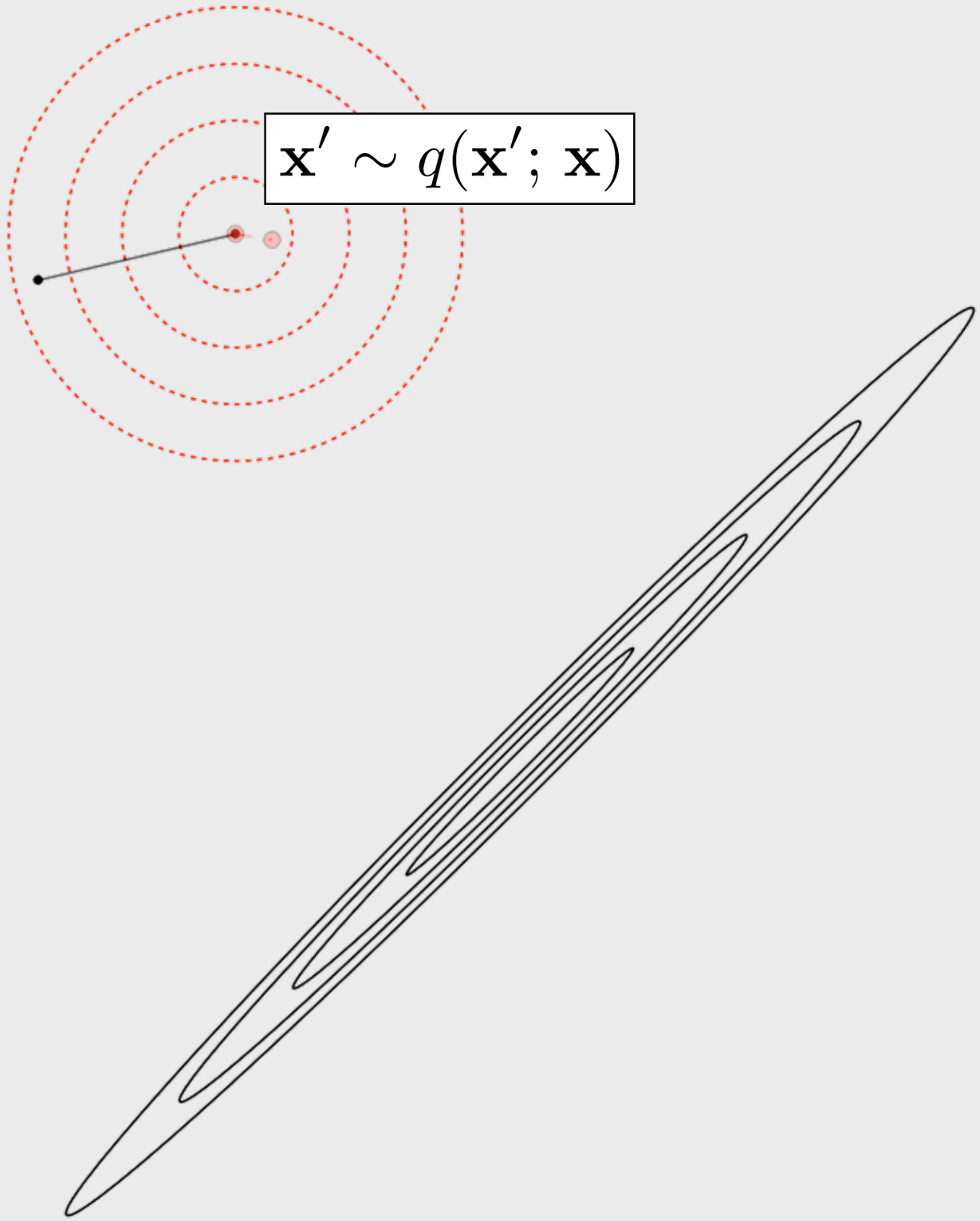
accept?

only **relative** probabilities

$$p(\text{accept}) = \min\left(1, \frac{p(\mathbf{x})}{p(\mathbf{x}')} \frac{q(\mathbf{x}; \mathbf{x}')}{q(\mathbf{x}'; \mathbf{x})}\right)$$

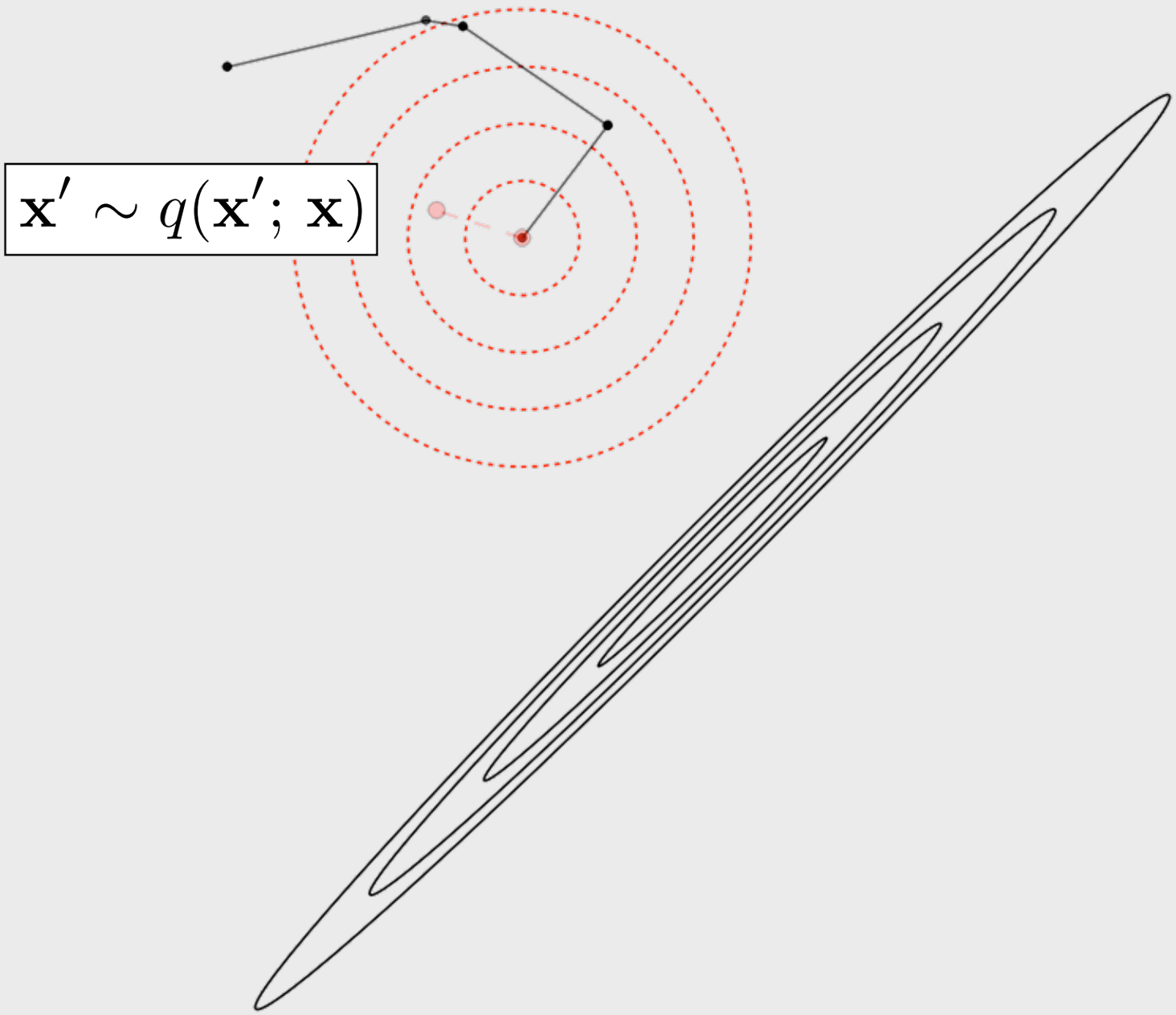
definitely.

Metropolis–Hastings
in an ideal world



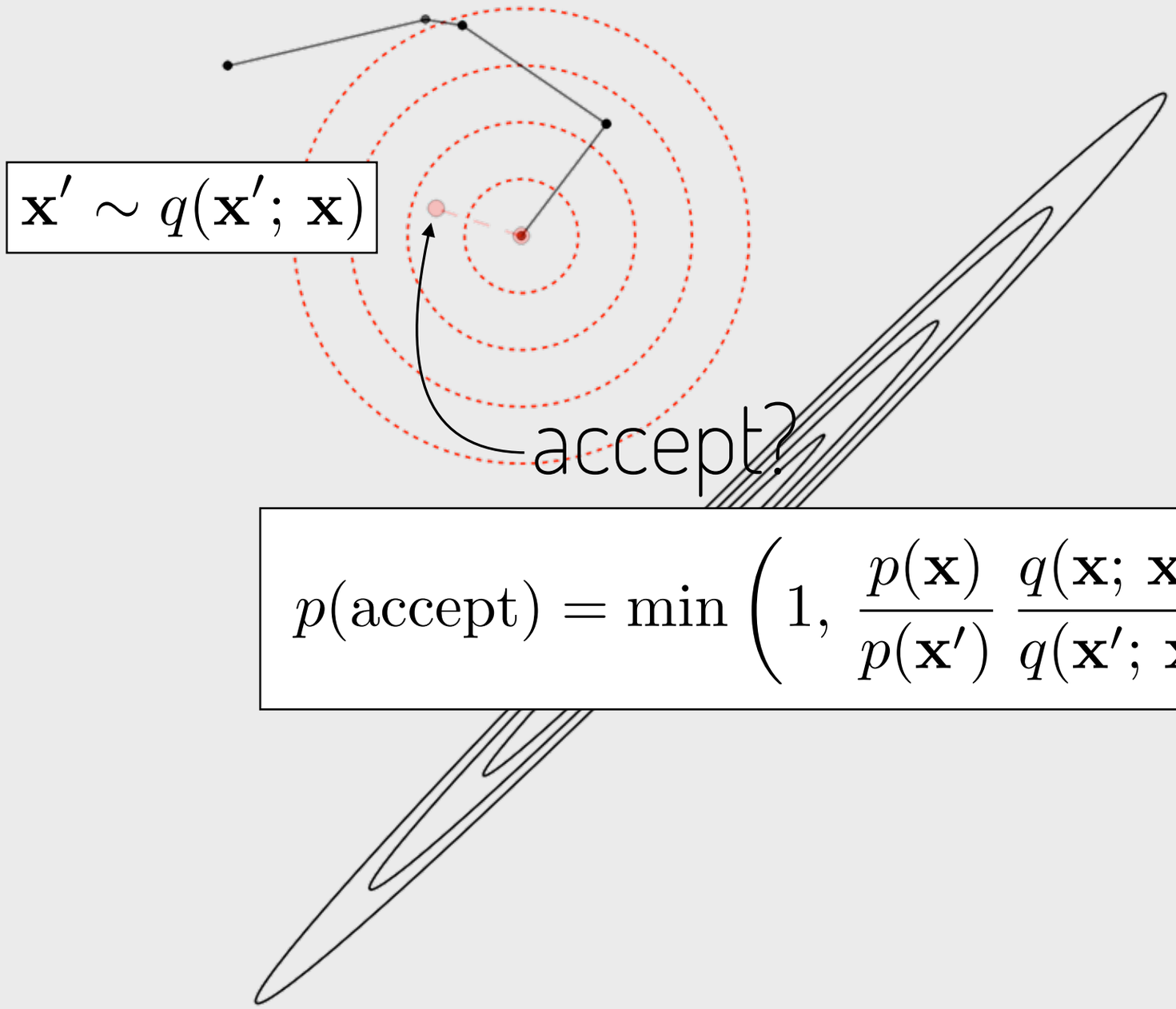
Metropolis–Hastings

in an ideal world



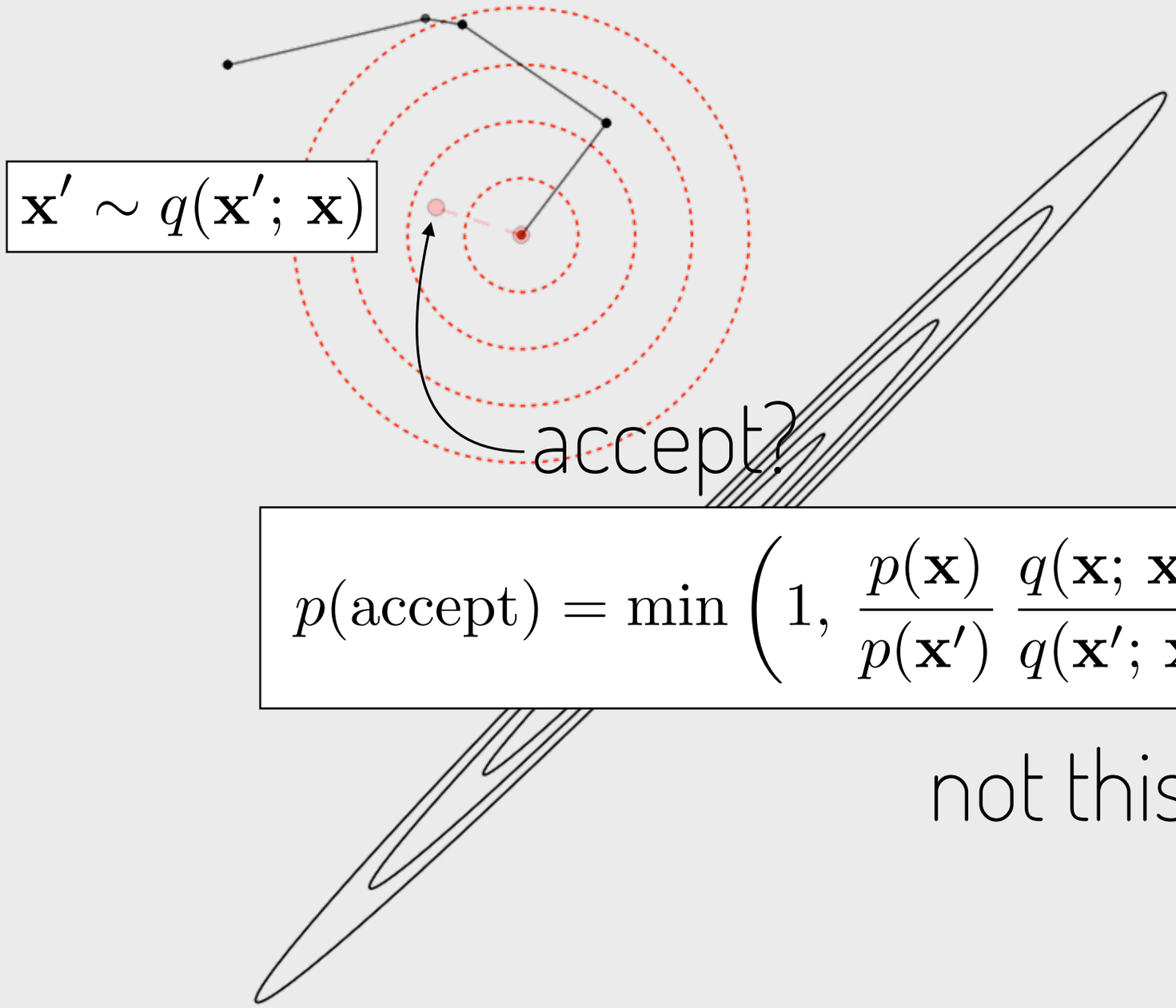
Metropolis–Hastings

in an ideal world



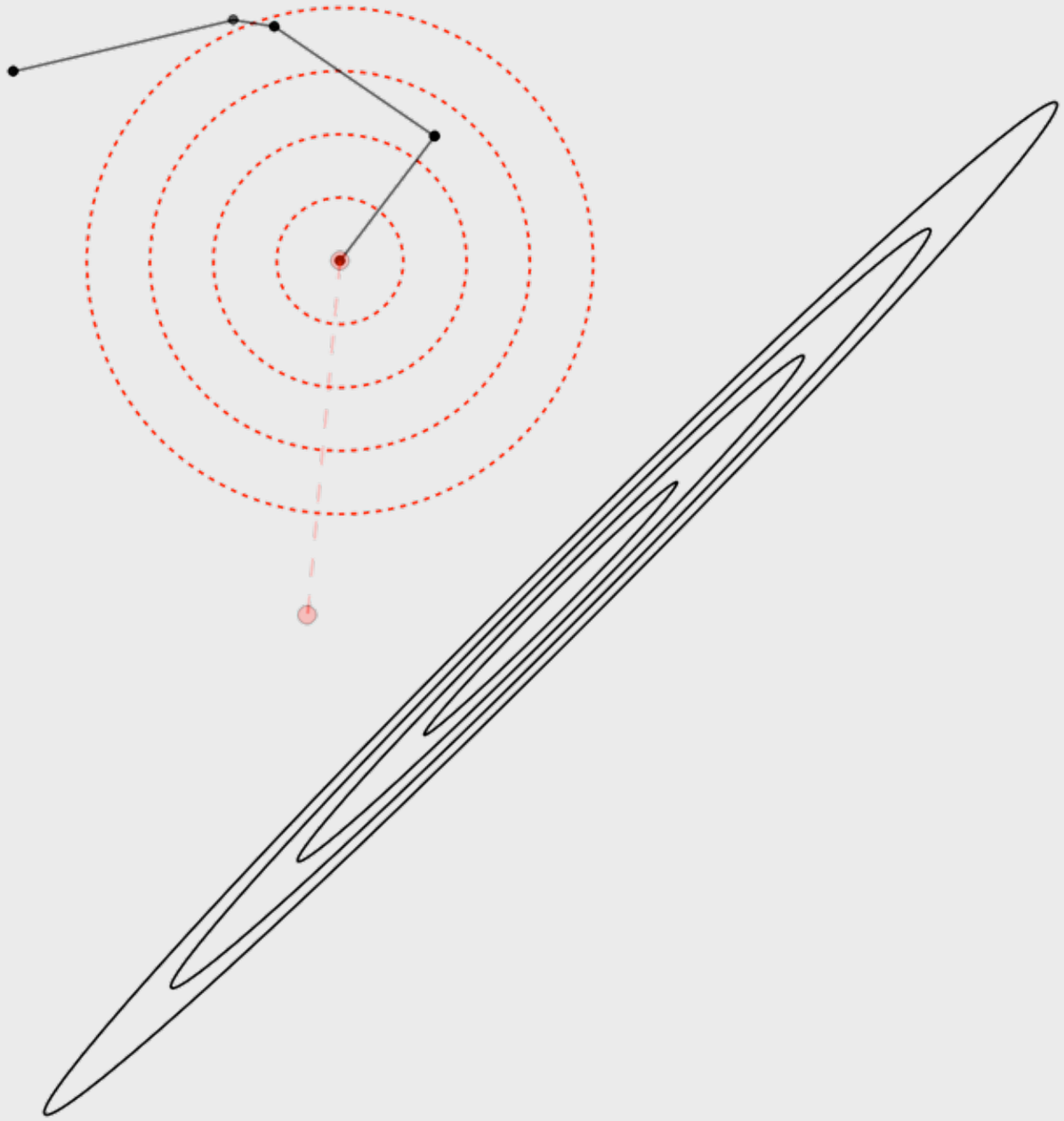
Metropolis–Hastings

in an ideal world



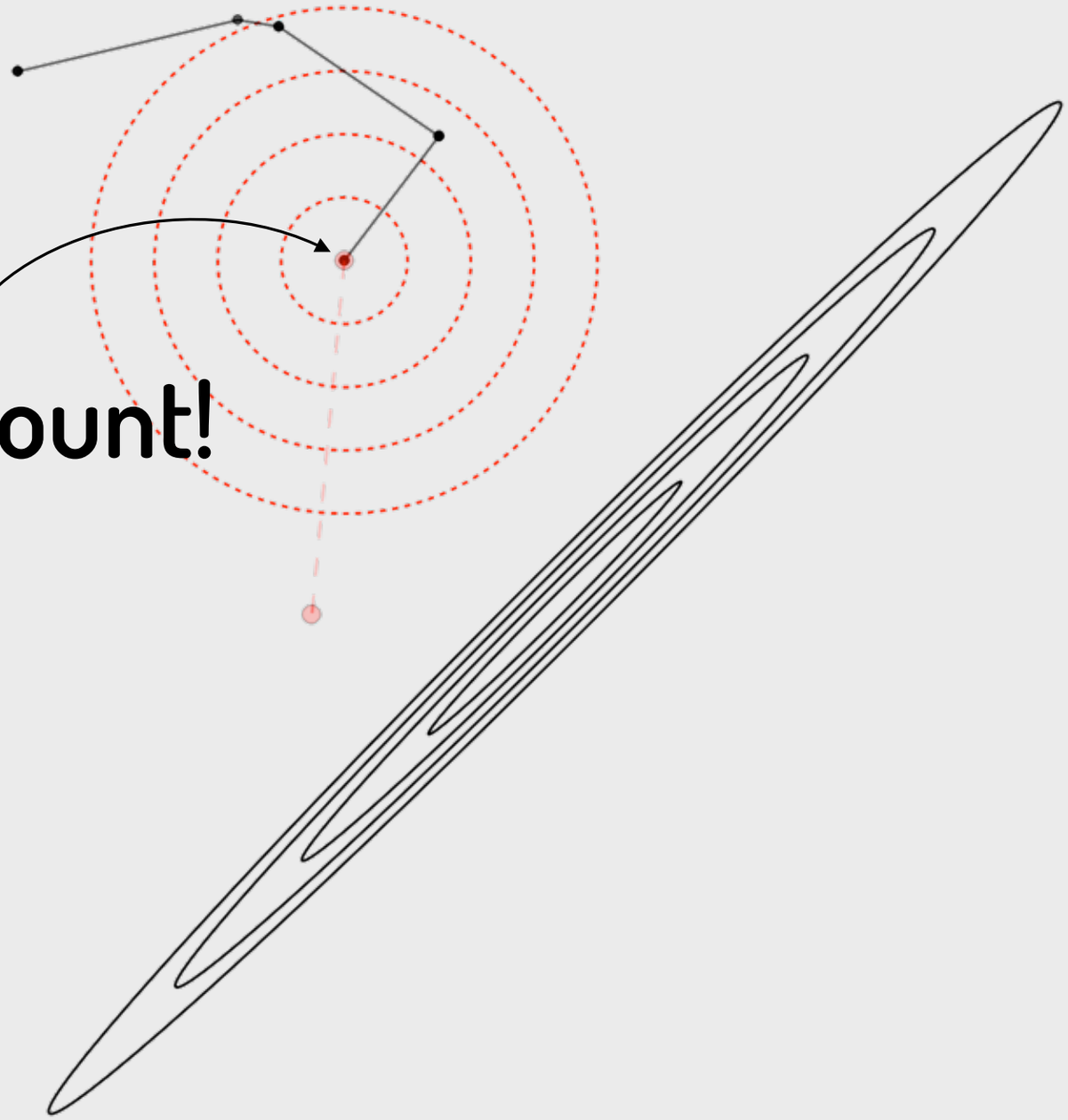
Metropolis–Hastings

in an ideal world



Metropolis–Hastings

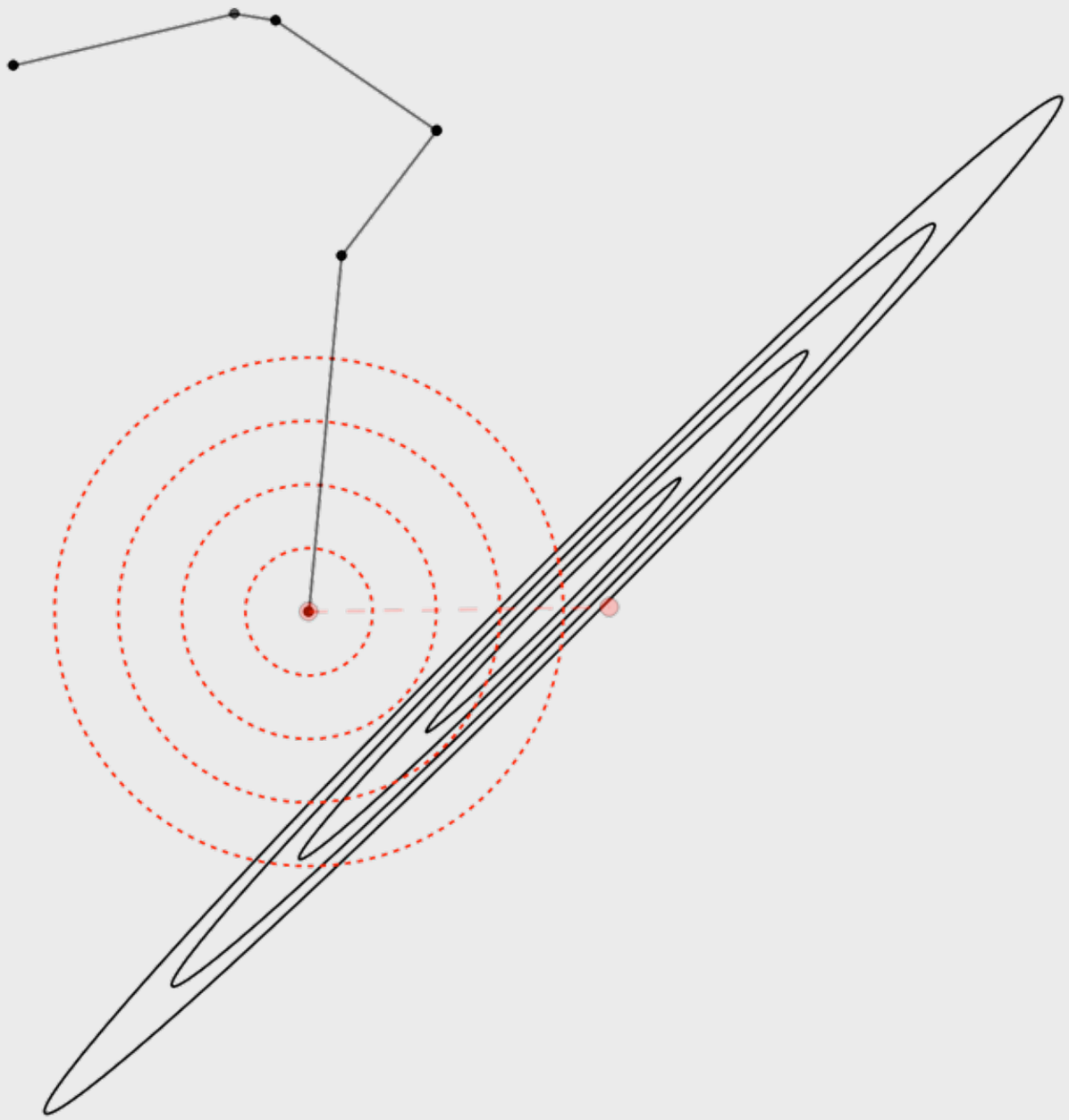
in an ideal world



double count!

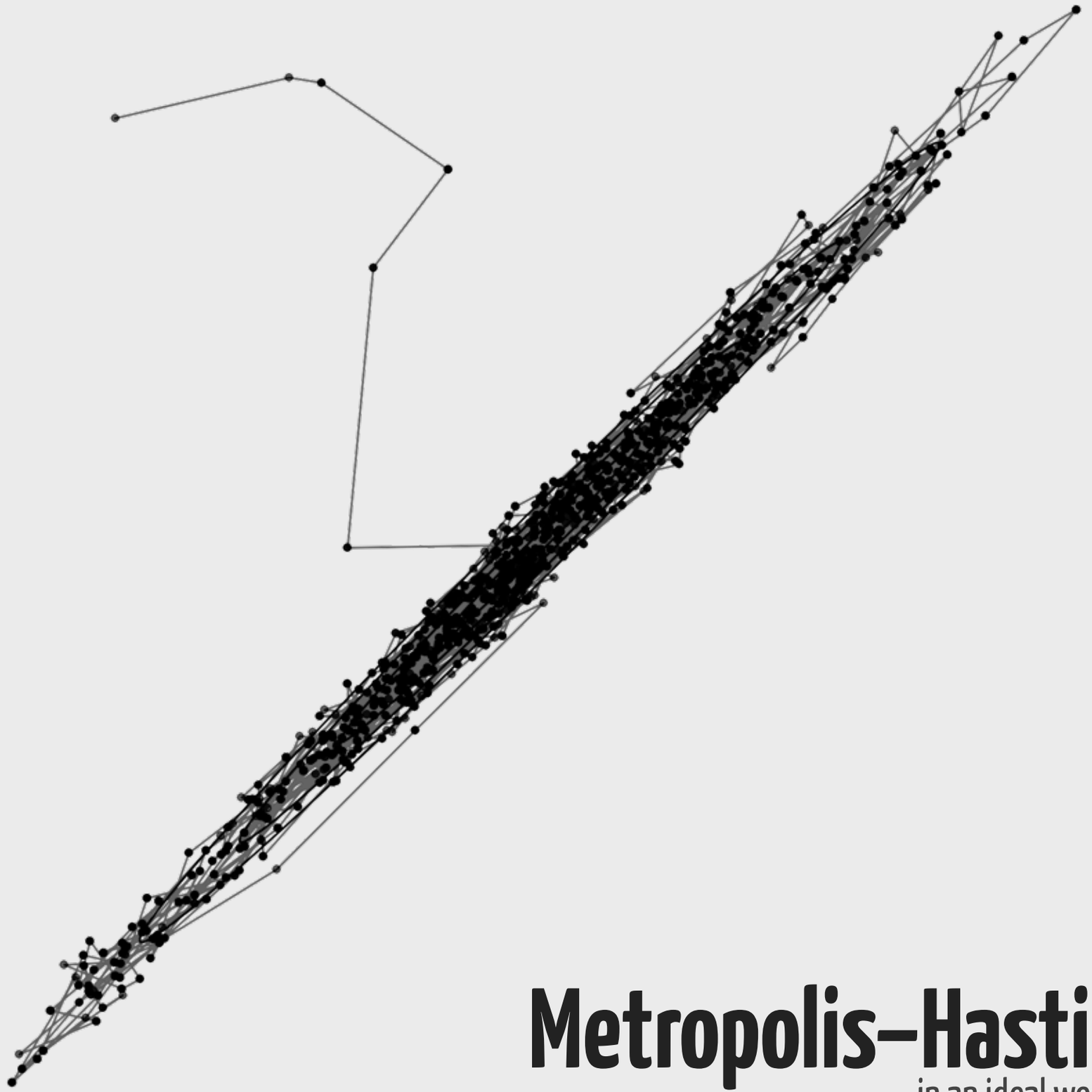
Metropolis–Hastings

in an ideal world



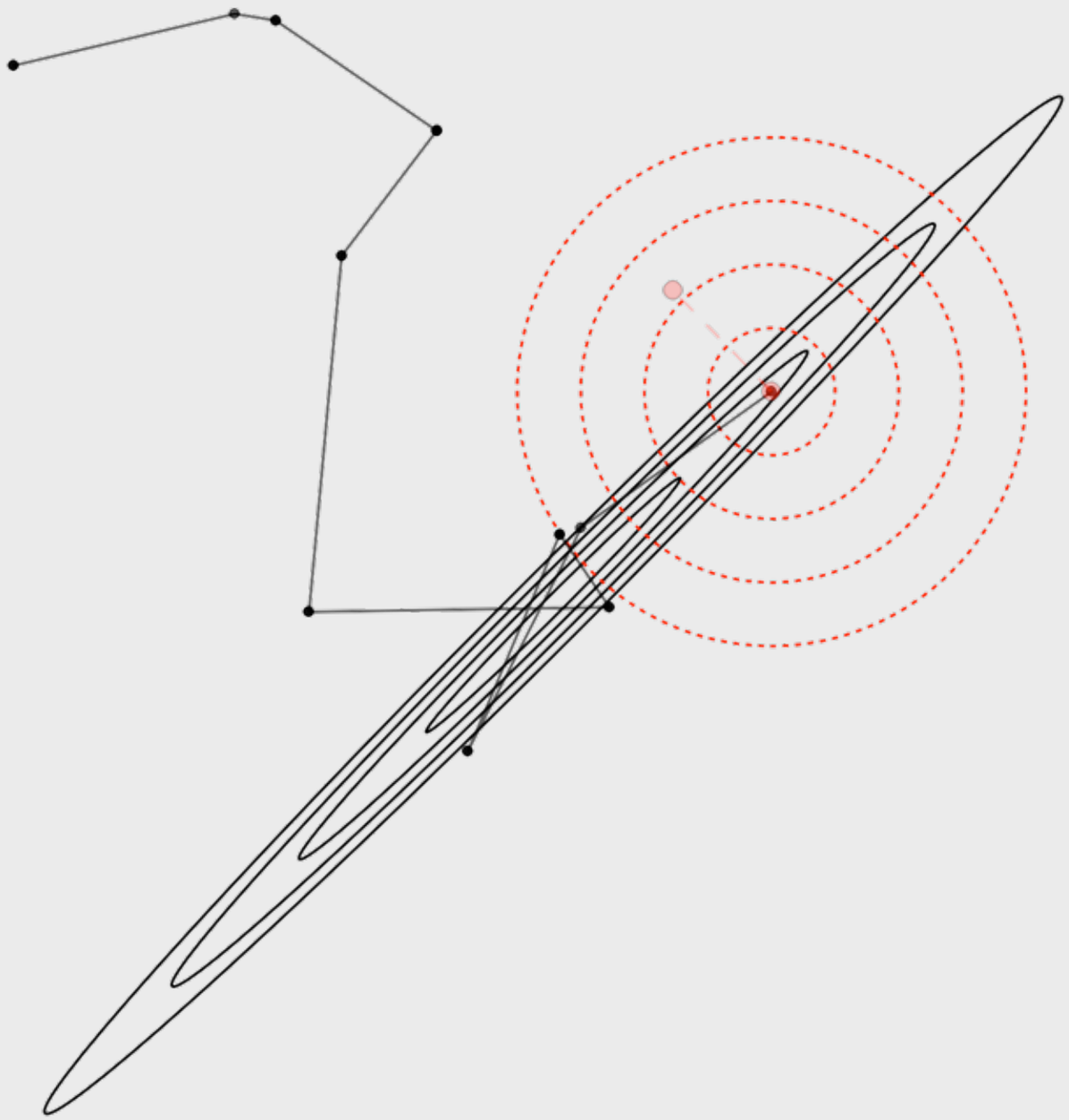
Metropolis–Hastings

in an ideal world



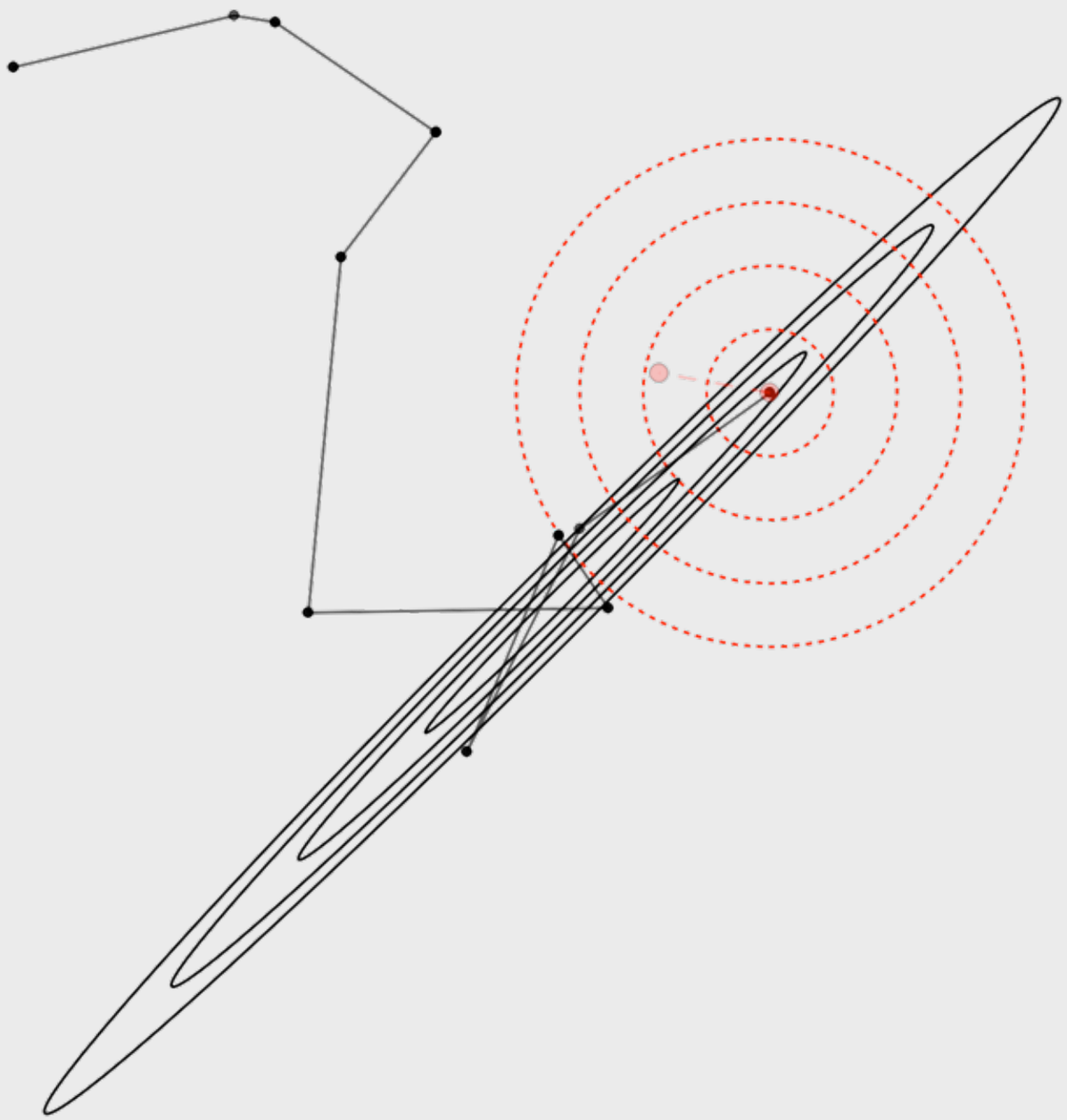
Metropolis–Hastings

in an ideal world



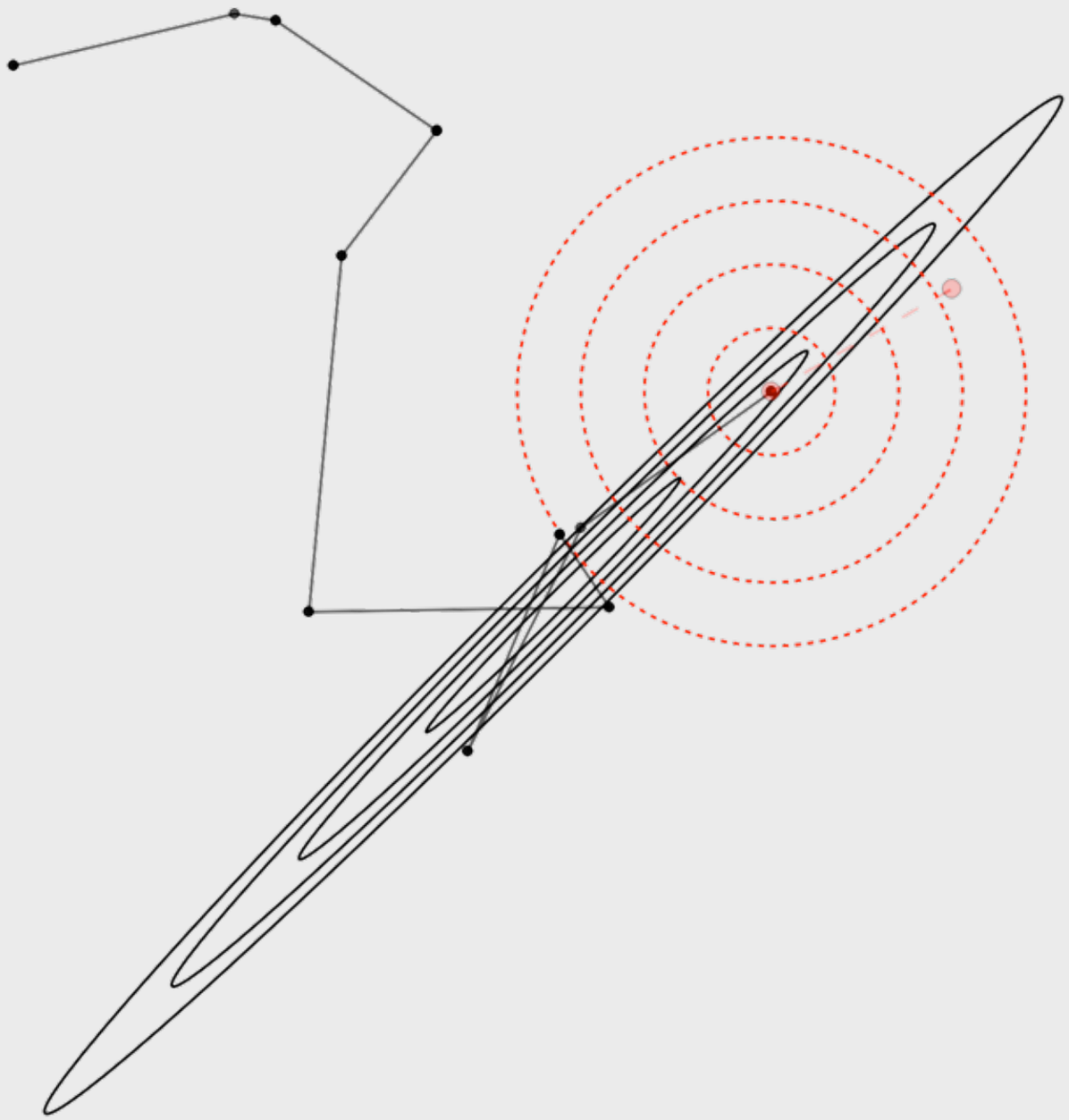
Metropolis–Hastings

in the real world



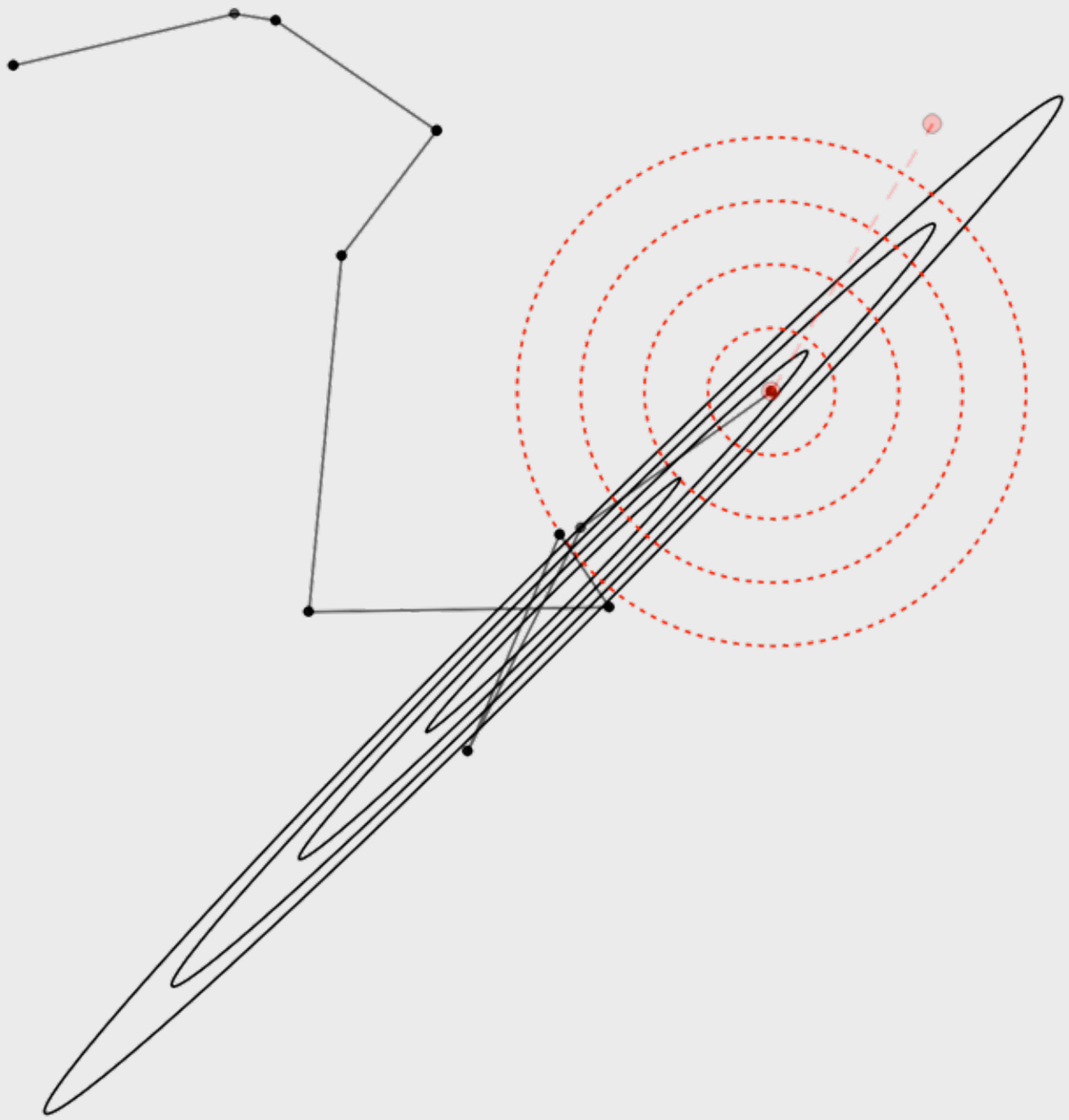
Metropolis–Hastings

in the real world



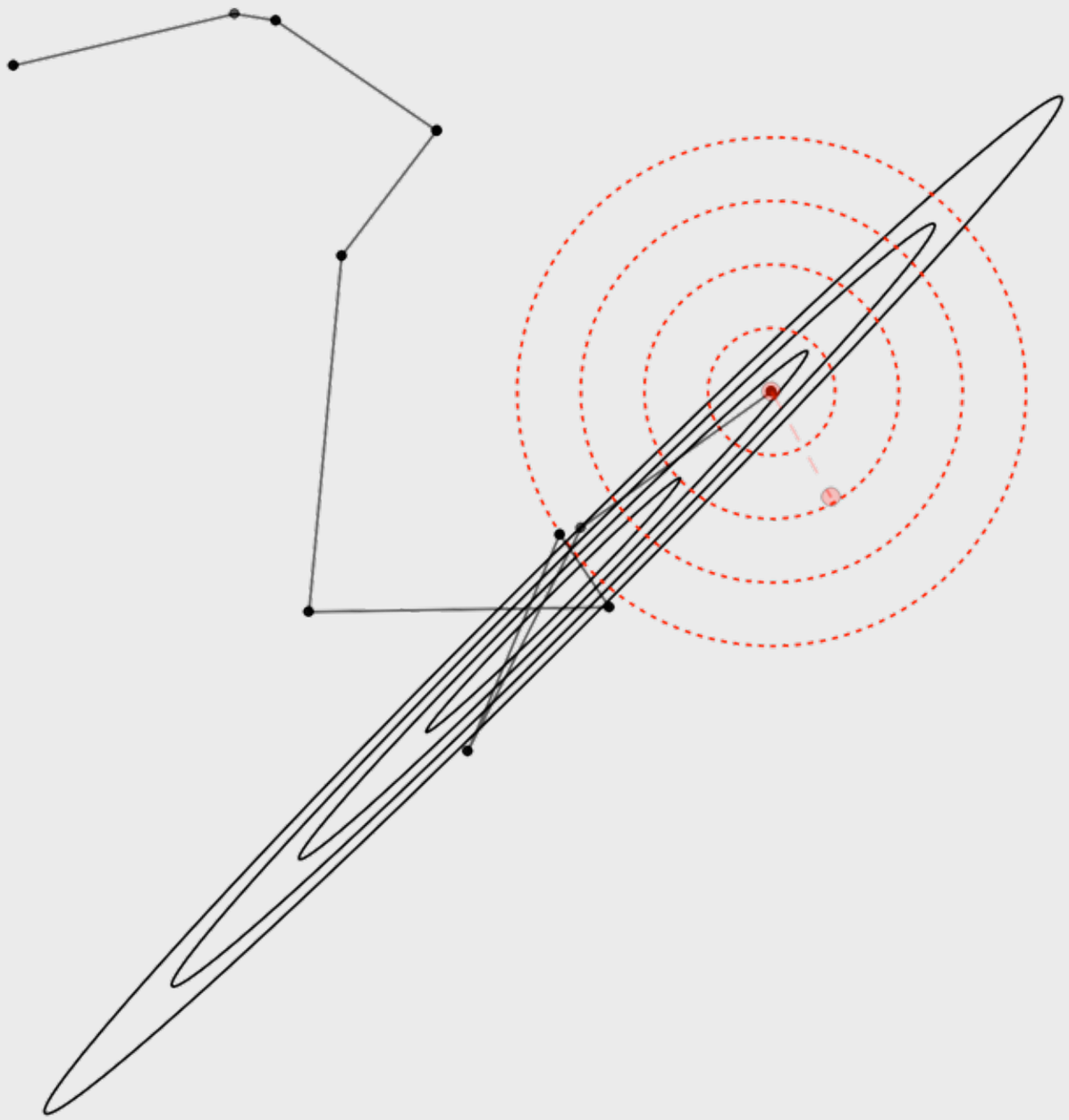
Metropolis–Hastings

in the real world



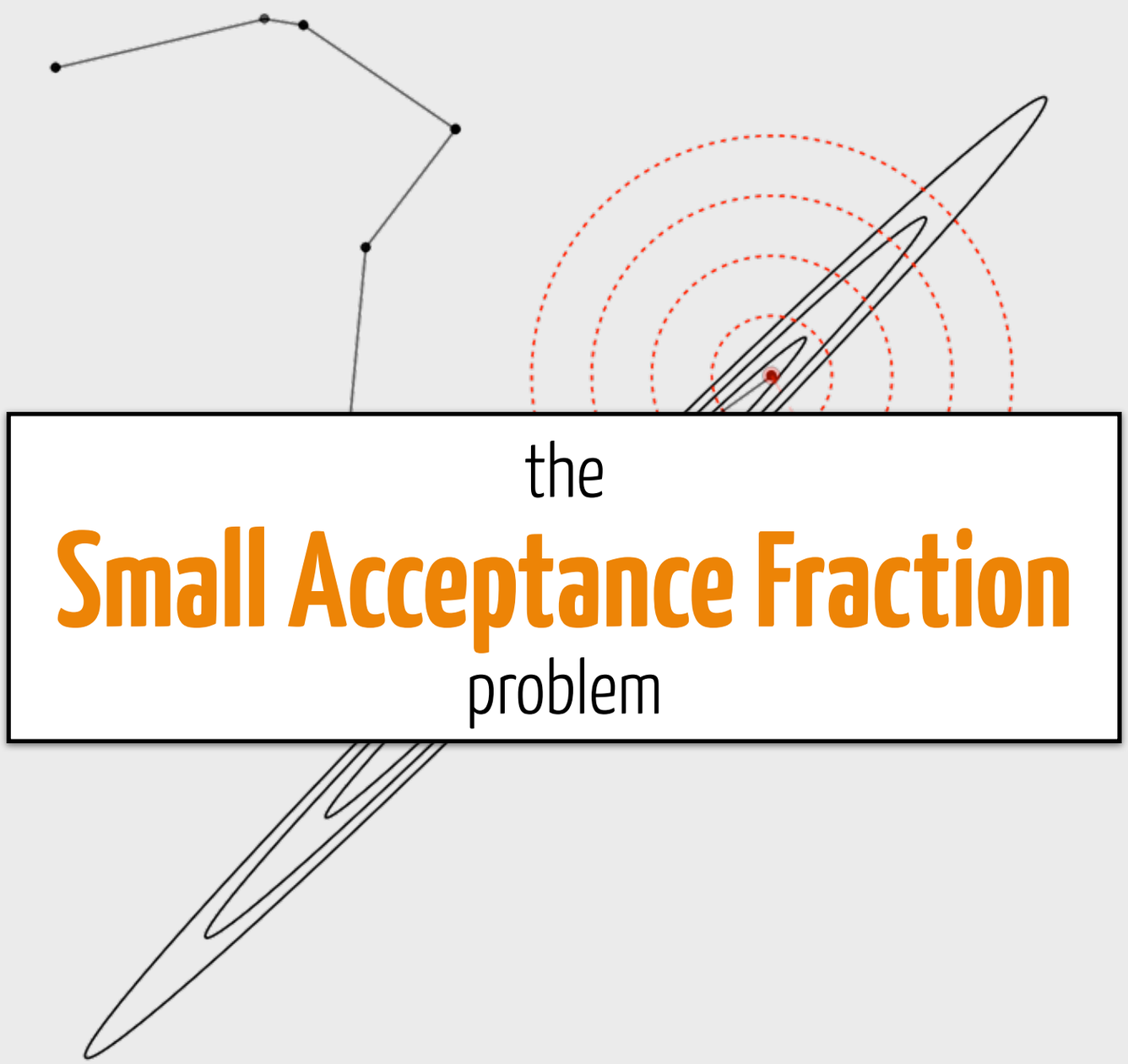
Metropolis–Hastings

in the real world



Metropolis–Hastings

in the real world



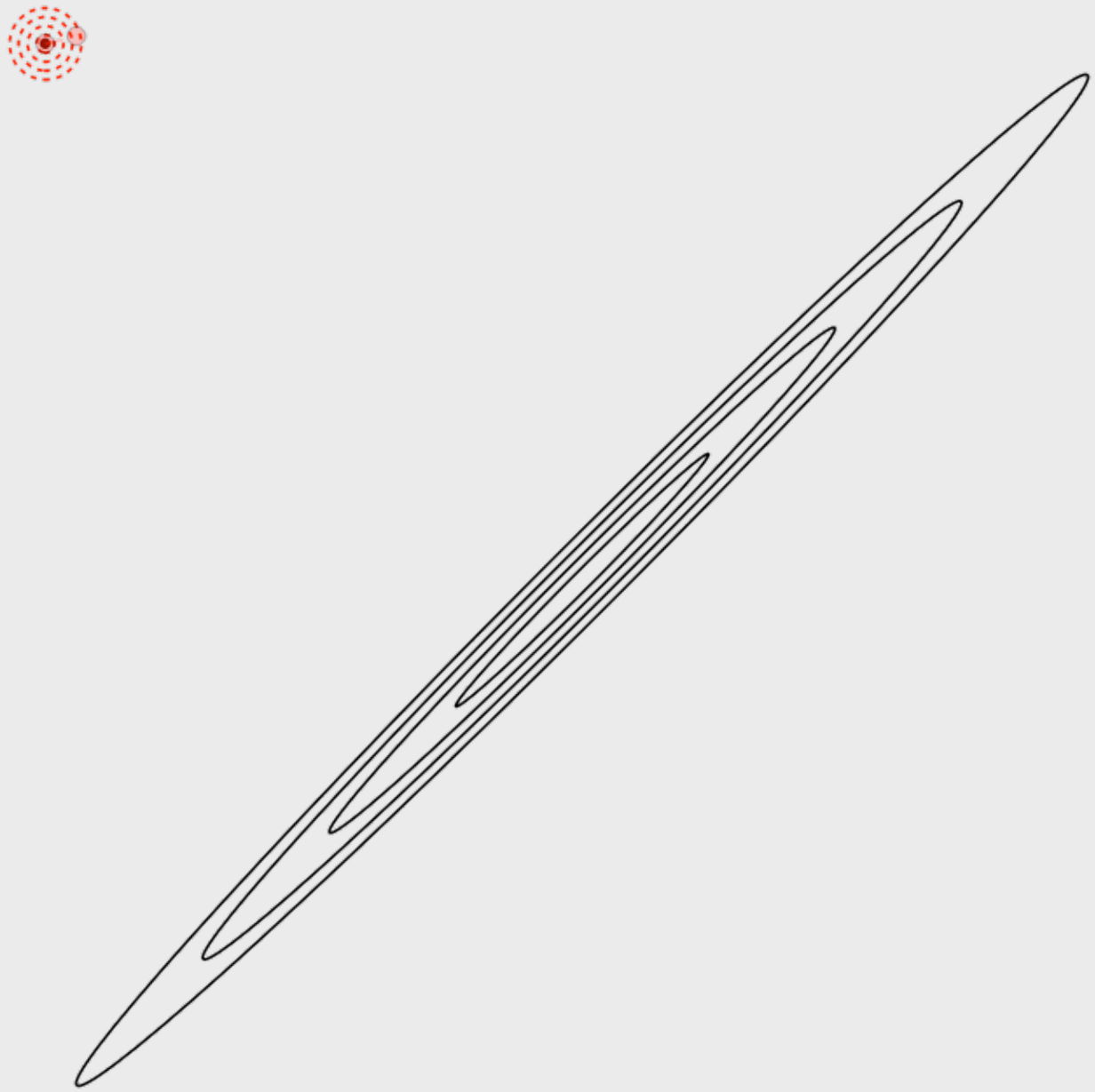
the

Small Acceptance Fraction

problem

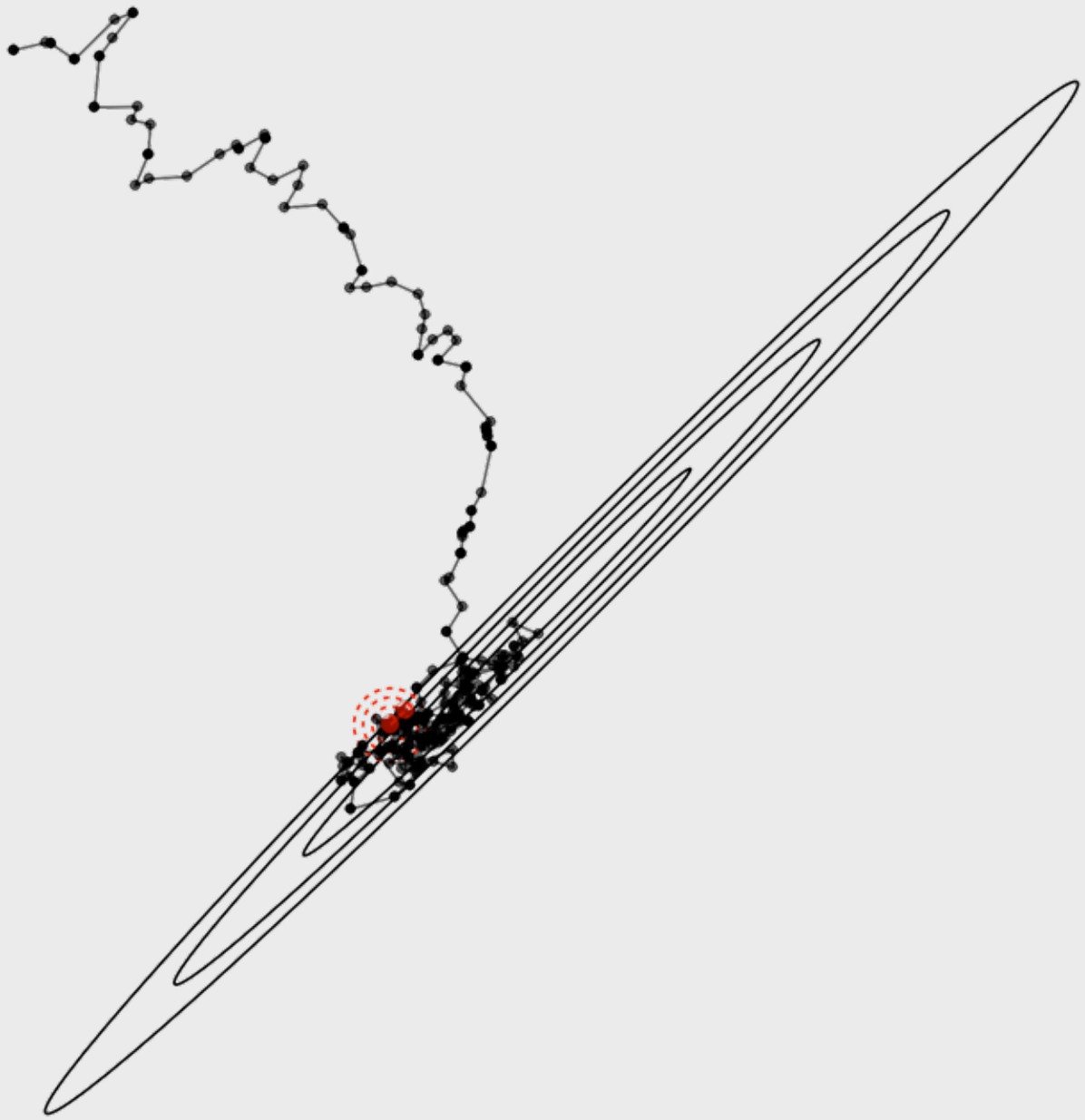
Metropolis–Hastings

in the real world



Metropolis–Hastings

in the real world



Metropolis–Hastings

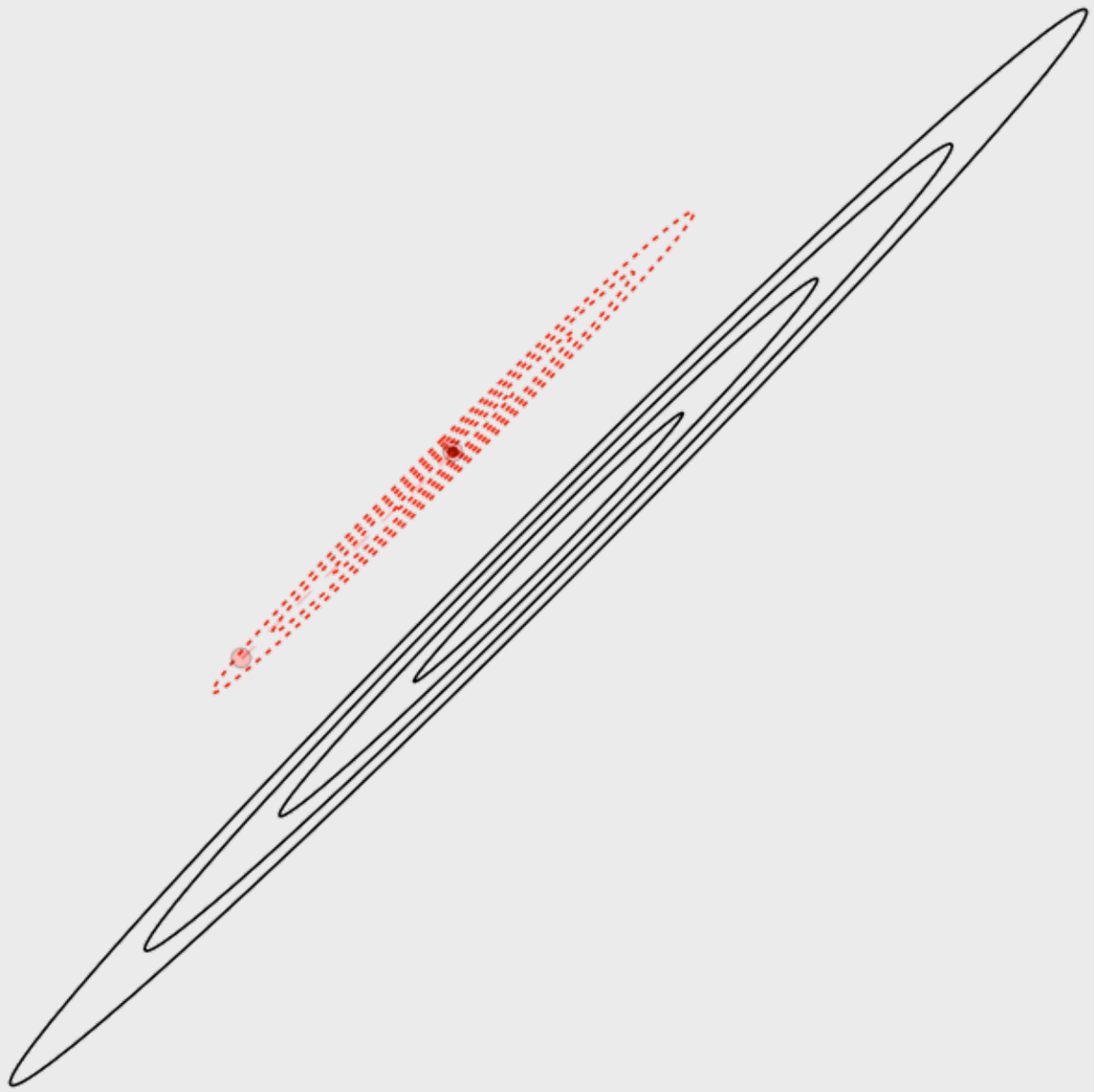
in the real world

Metropolis–Hastings

in the real world

a brief aside.

YOUR LIKELIHOOD FUNCTION IS EXPENSIVE

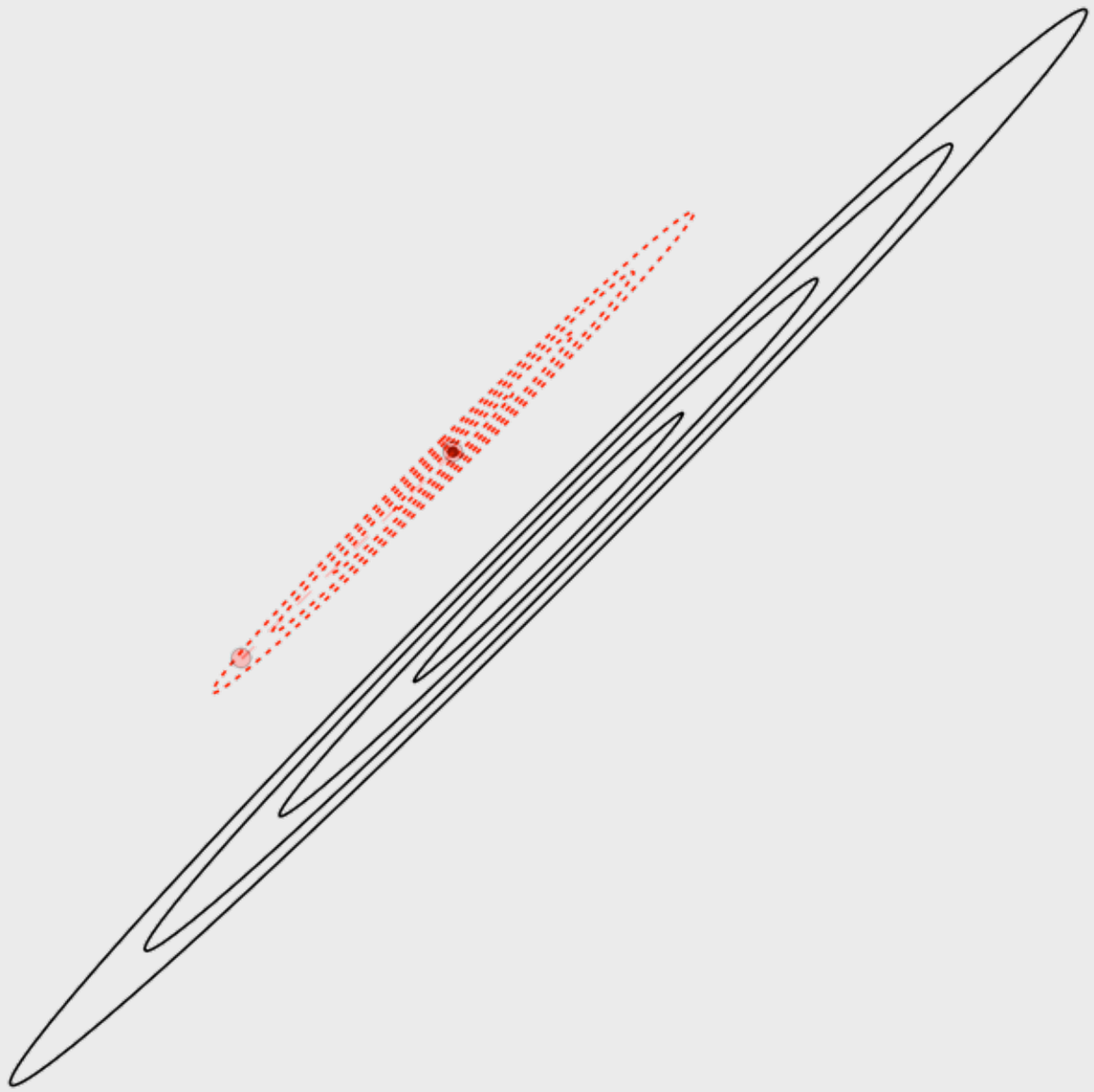


Metropolis–Hastings

in the real world

REMEMBER?

YOUR LIKELIHOOD FUNCTION IS EXPENSIVE

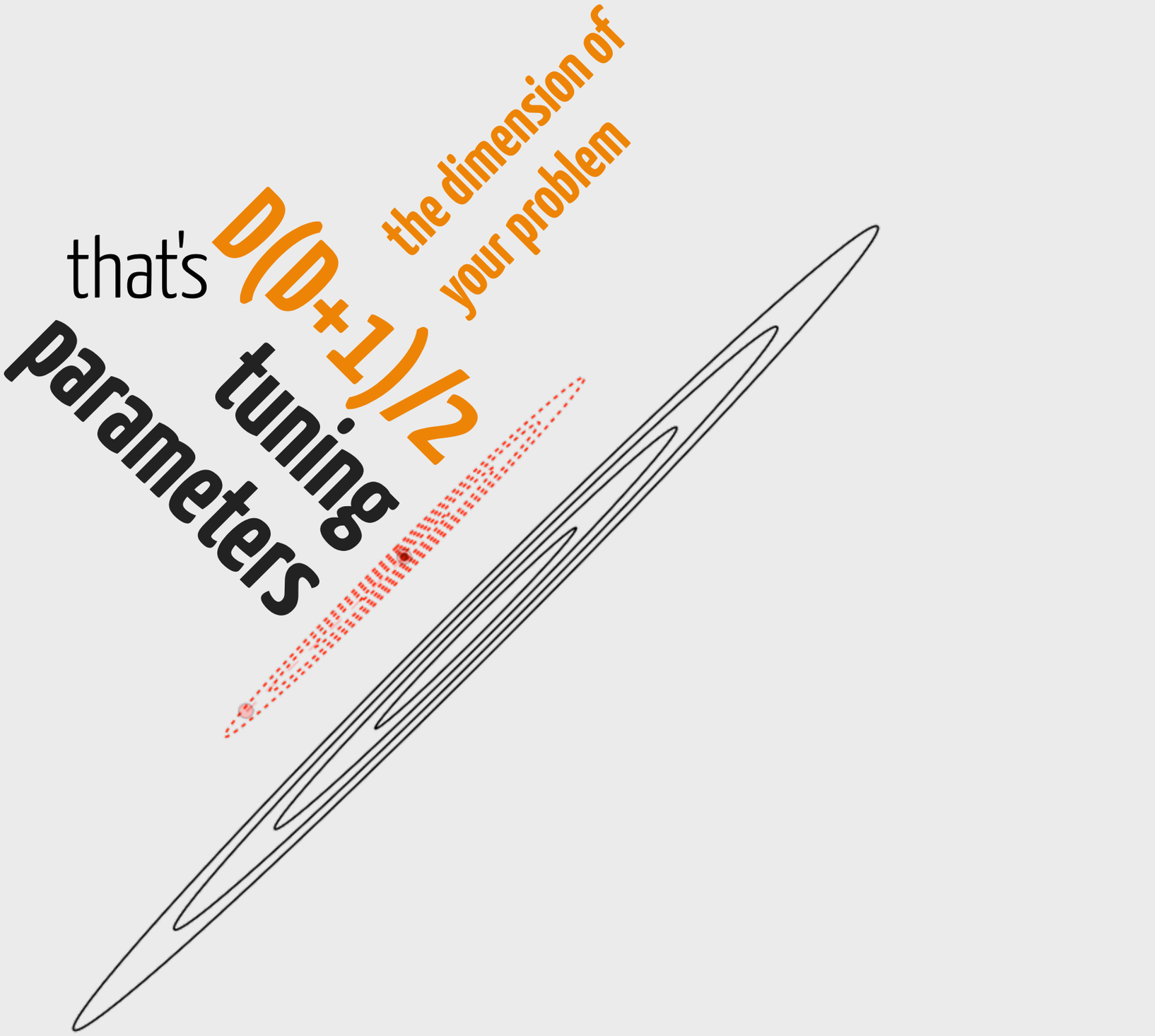


Metropolis–Hastings

in the real world

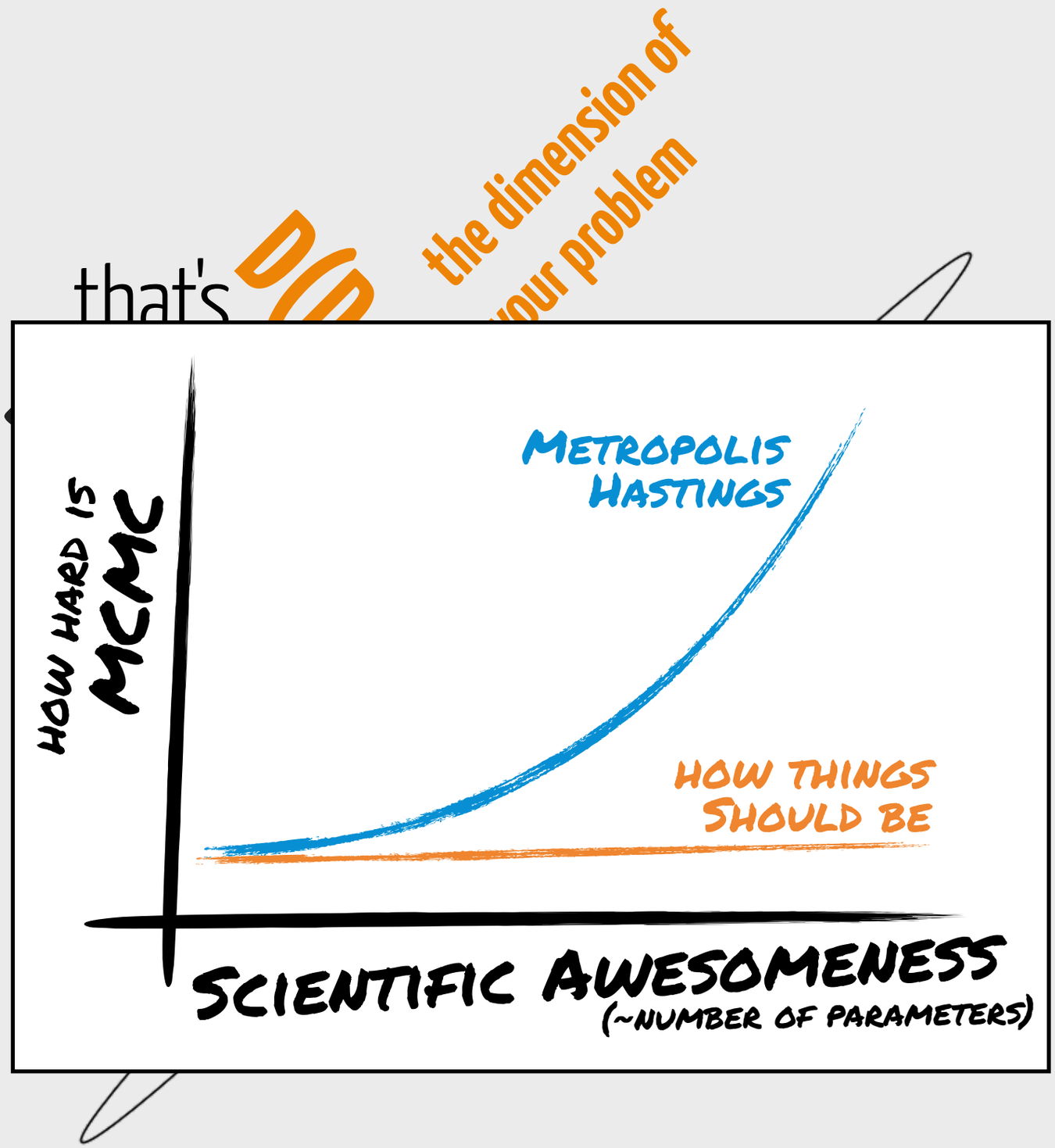
Metropolis–Hastings

in the real world



Metropolis–Hastings

in the real world



Metropolis–Hastings
in the real world

that being said,

go code up your own

Metropolis–Hastings

code right now.

(well not right right now)

Jonathan Goodman

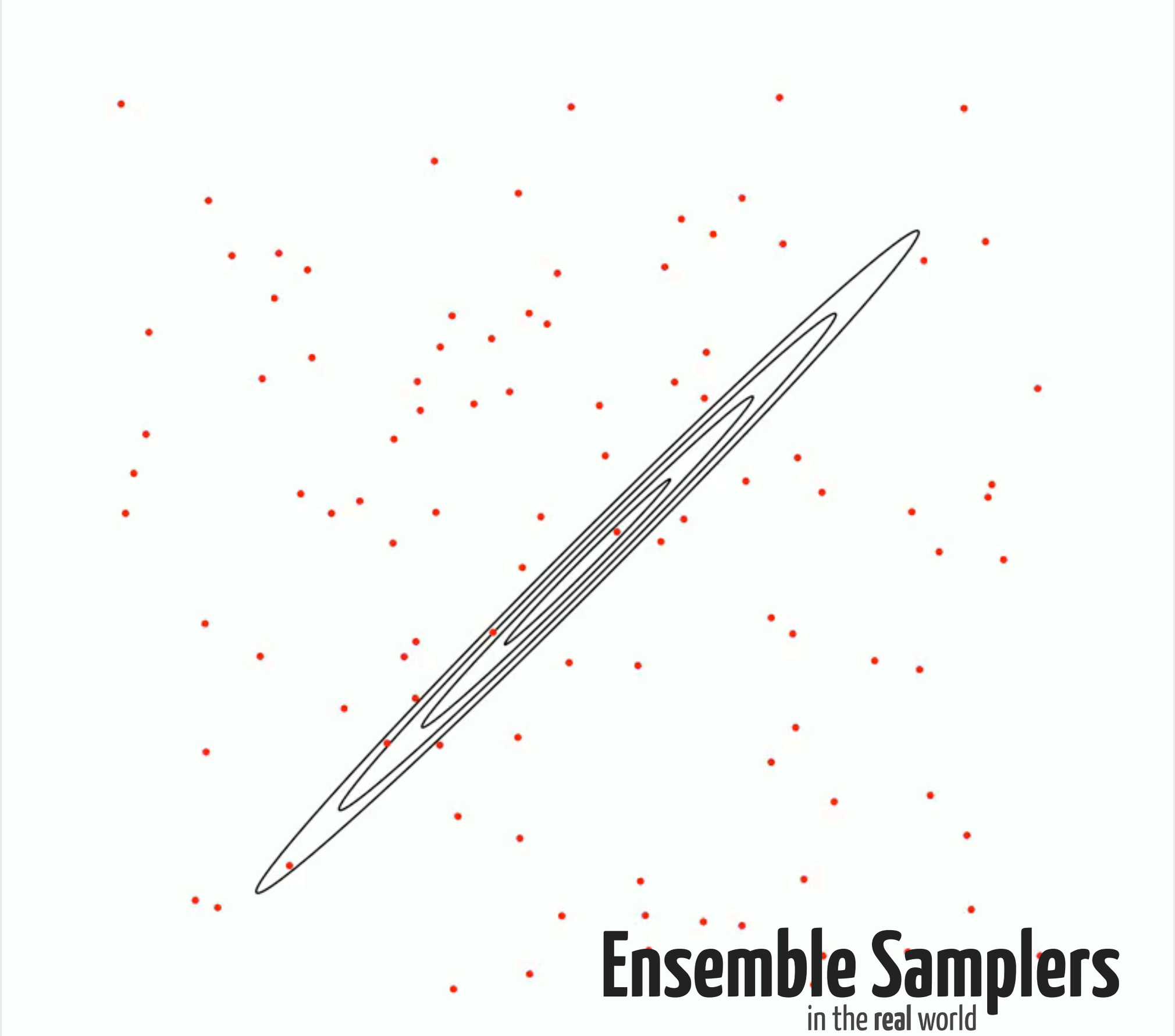
Jonathan Weare

"Ensemble samplers with affine invariance"

(dfm.io/mcmc-gw10)

Ensemble Samplers

in the **real** world



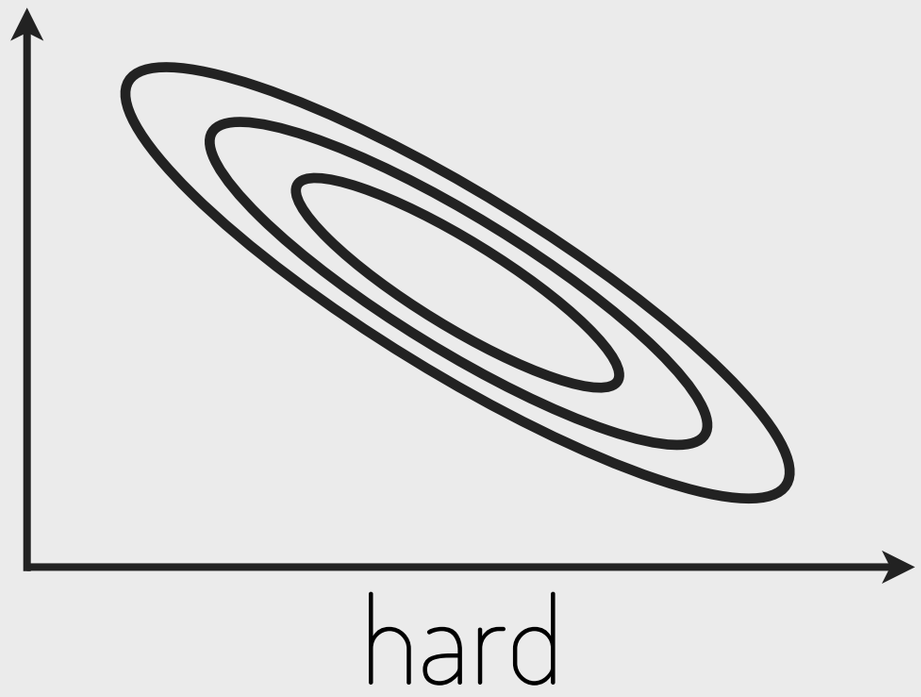
Ensemble Samplers

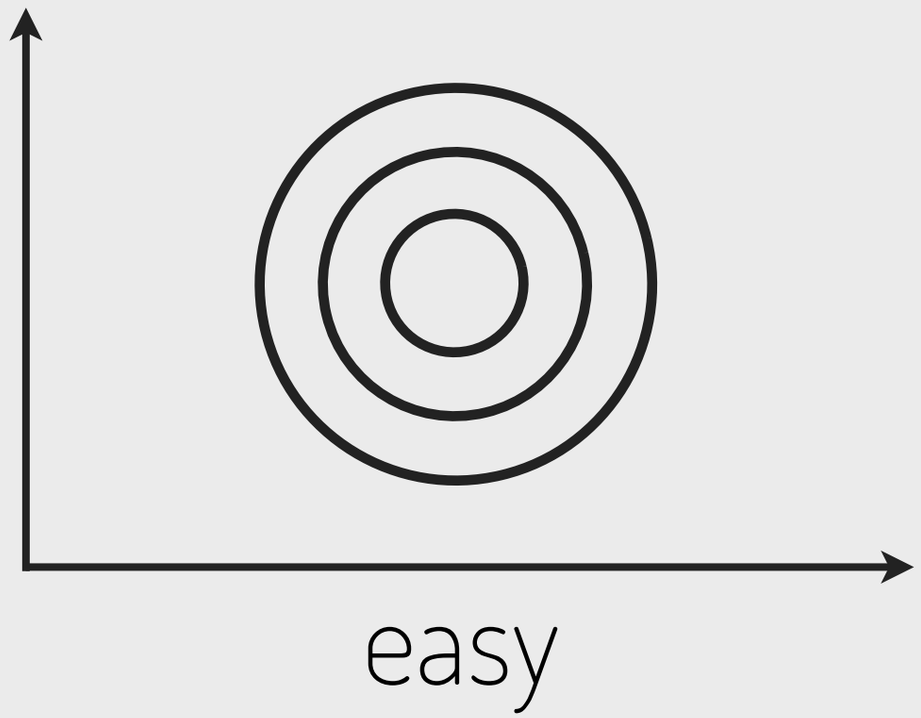
in the **real** world

Ensemble samplers with affine invariance

Affine Transformation

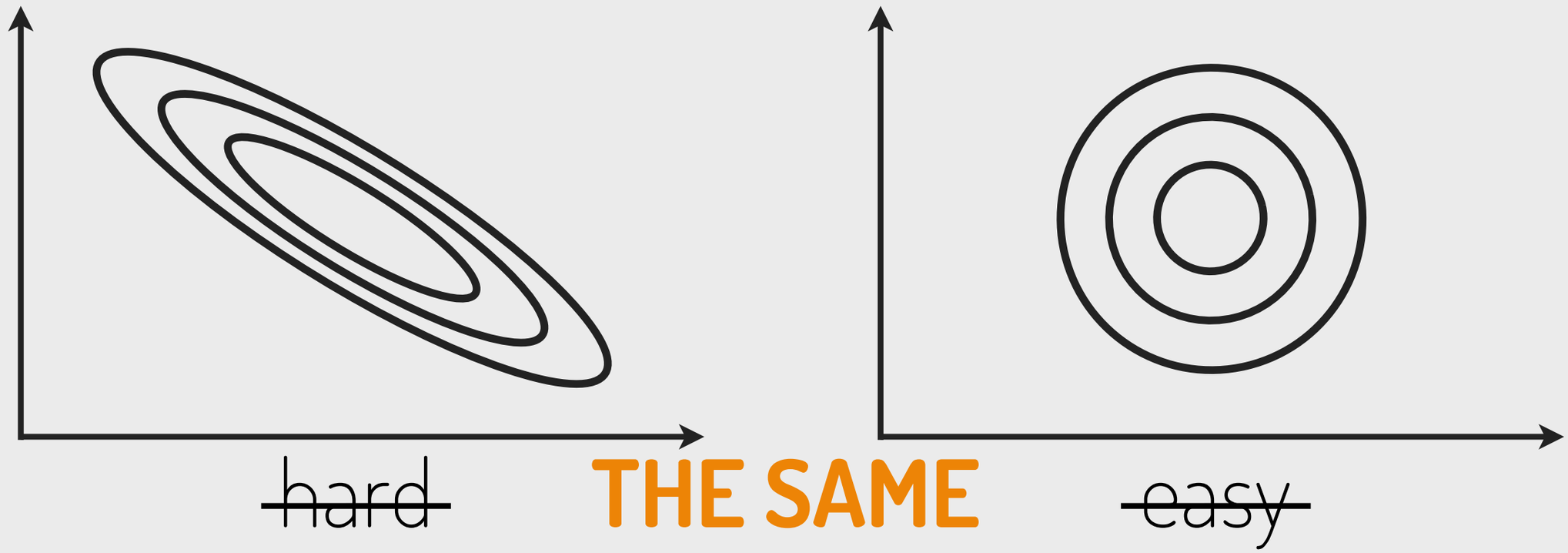
$$y \leftarrow A\,x + b$$

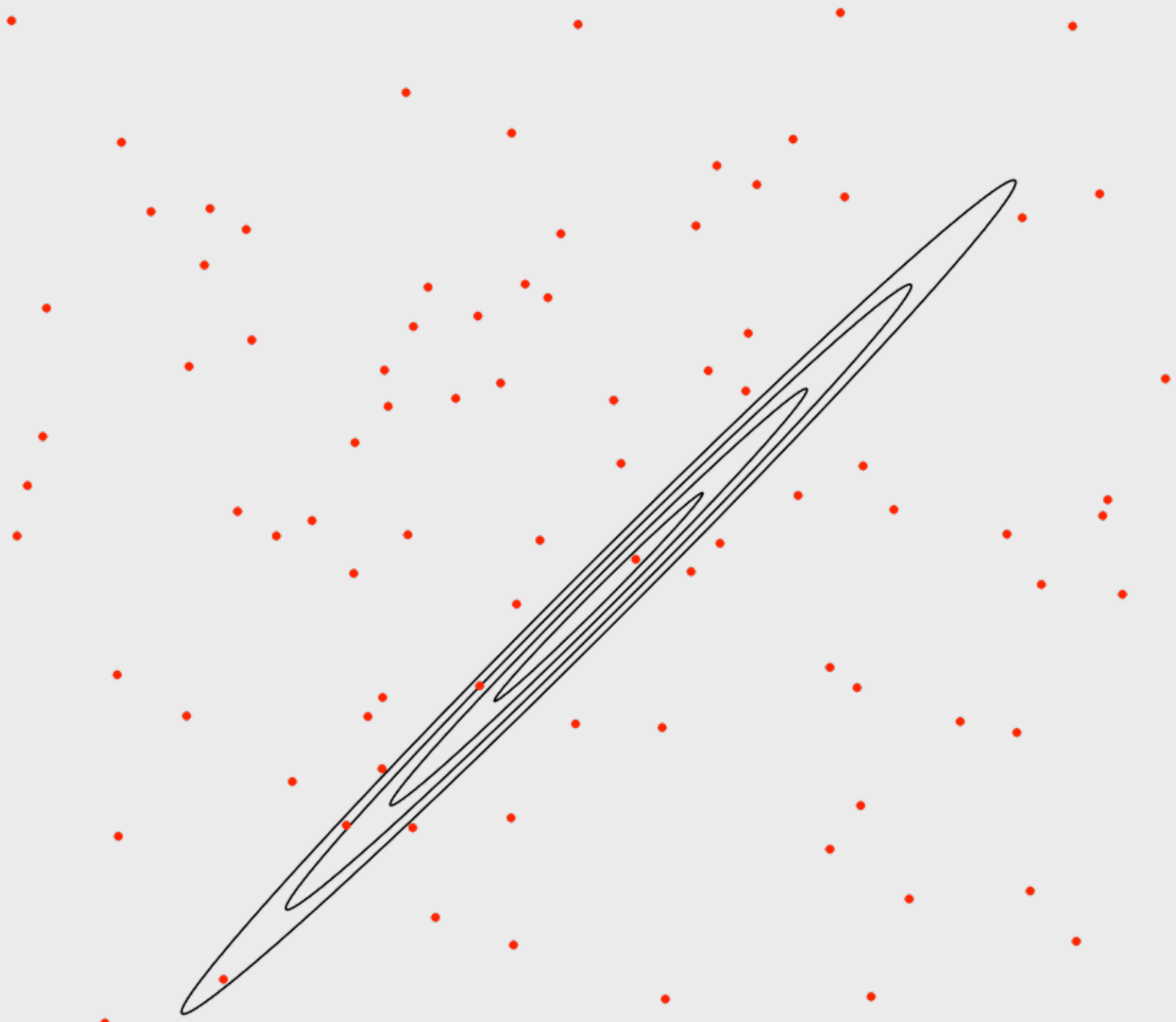




Affine Transformation

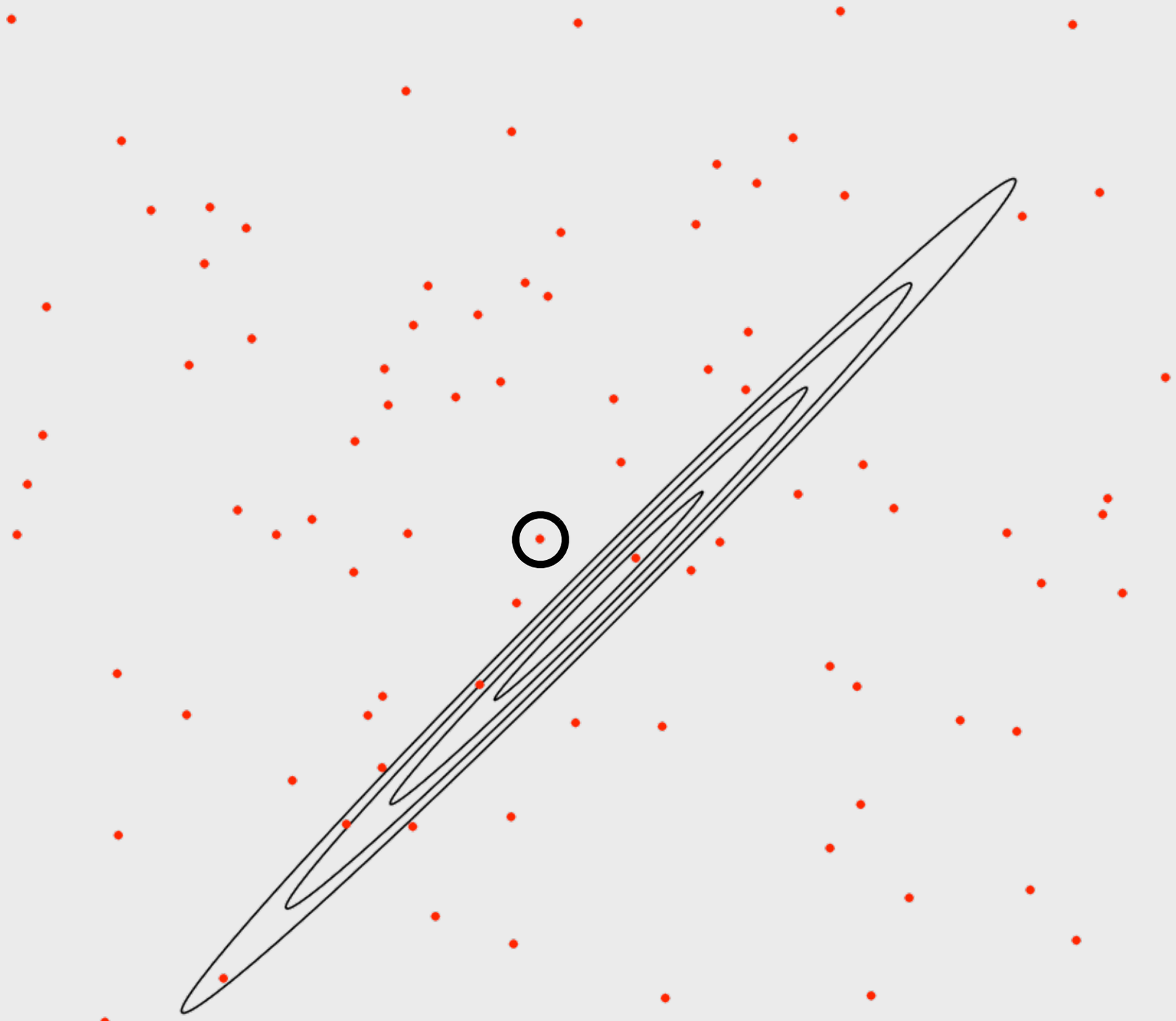
$$y \leftarrow A\,x + b$$





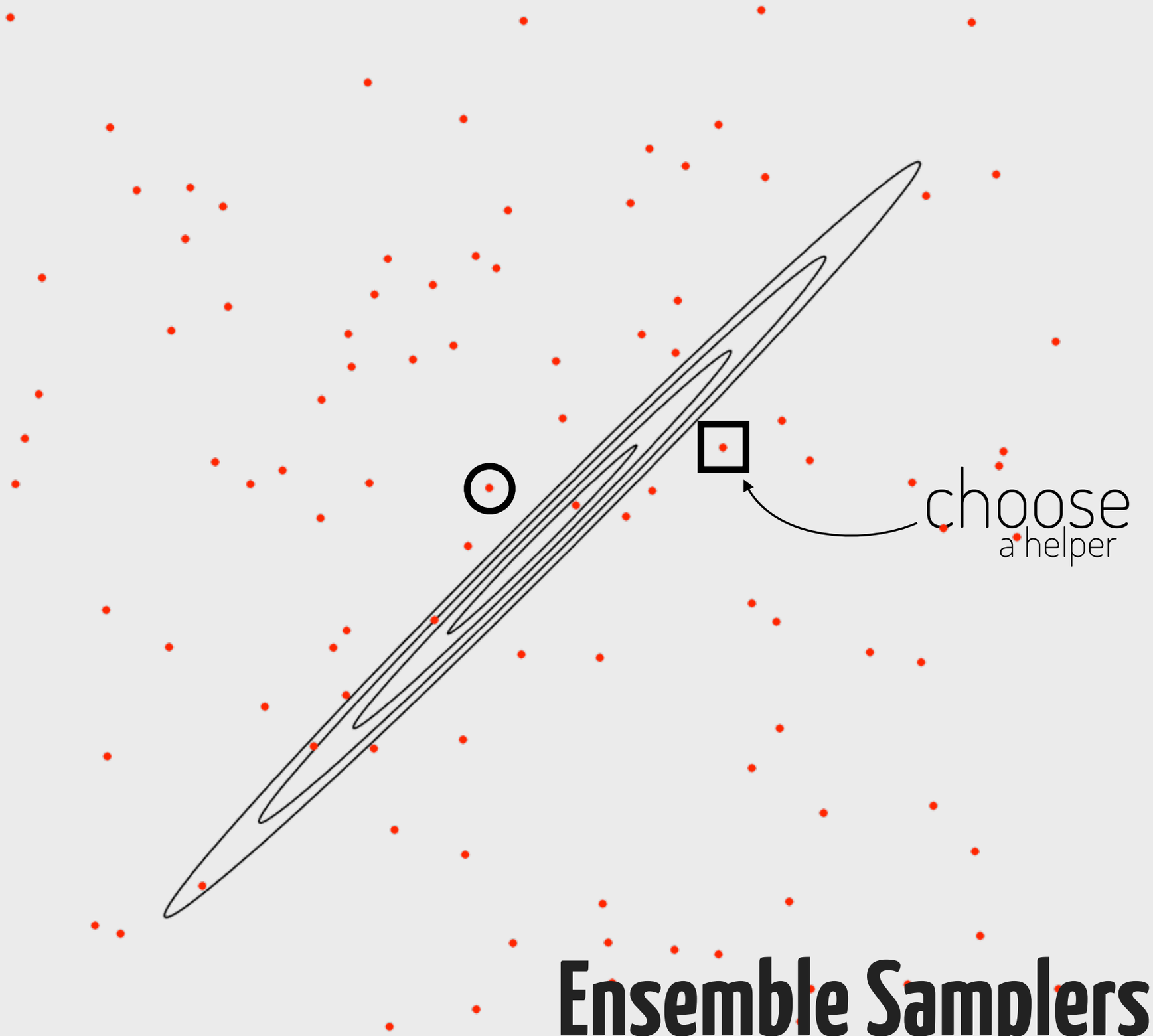
Ensemble Samplers

in the **real** world



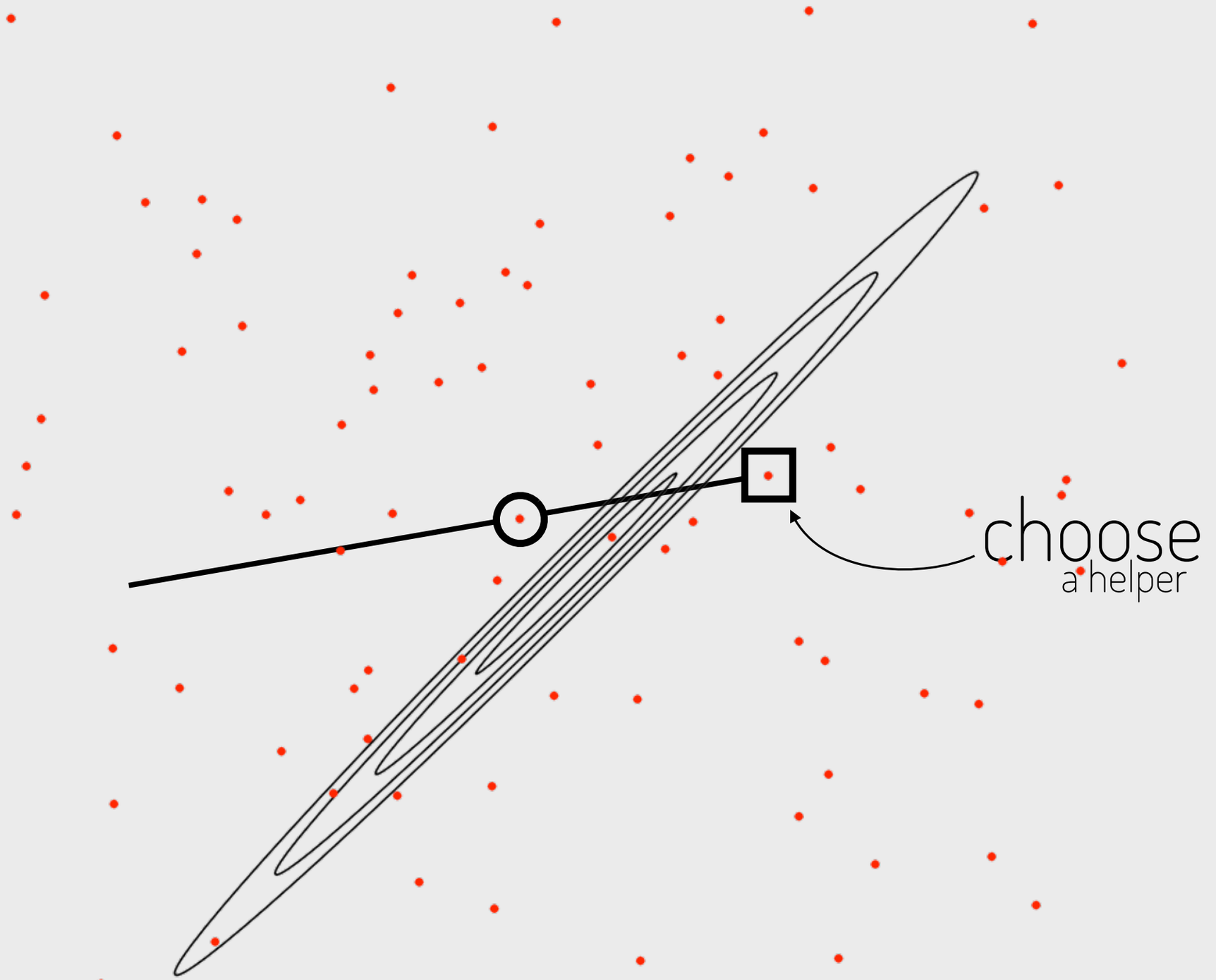
Ensemble Samplers

in the **real** world



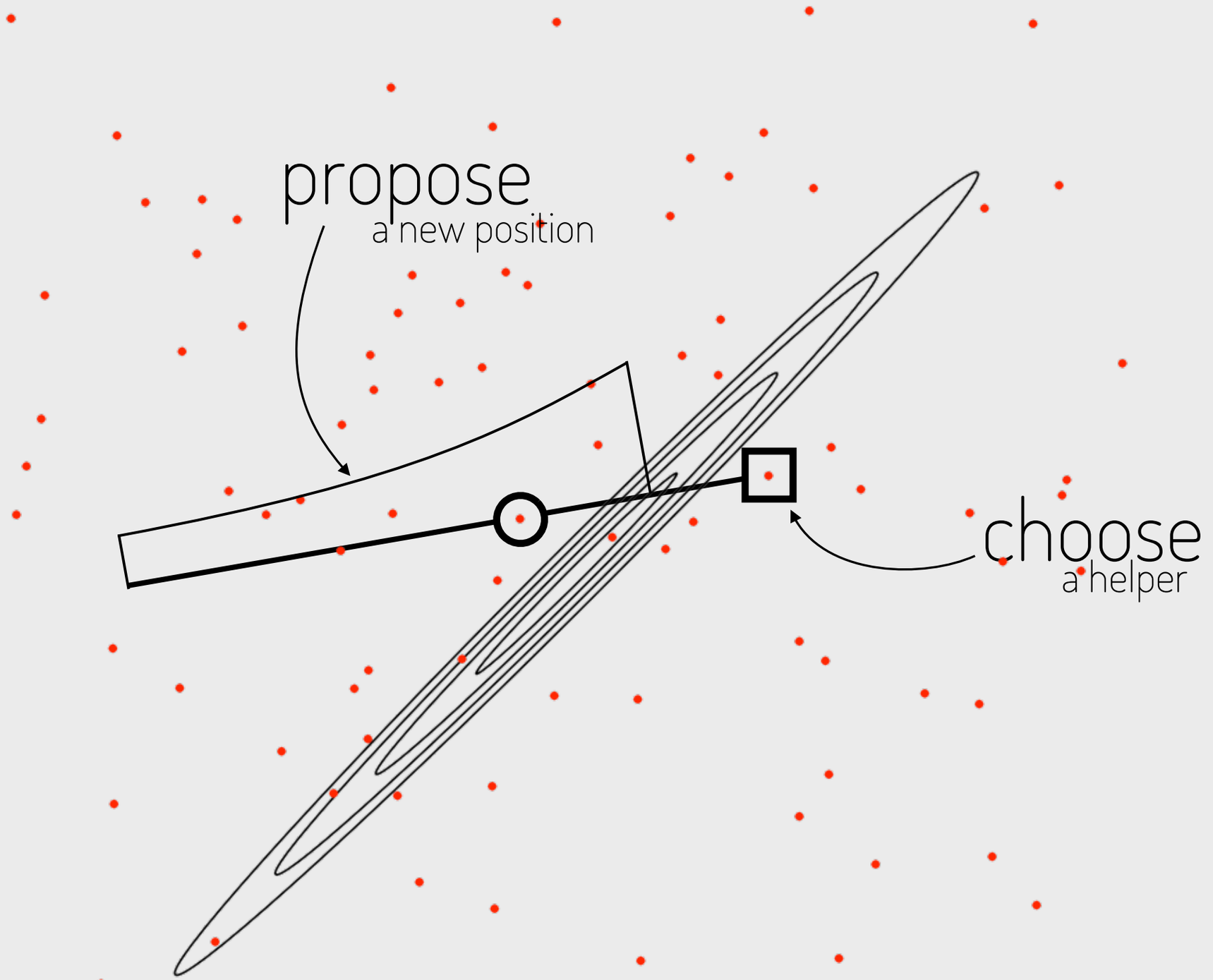
Ensemble Samplers

in the **real** world



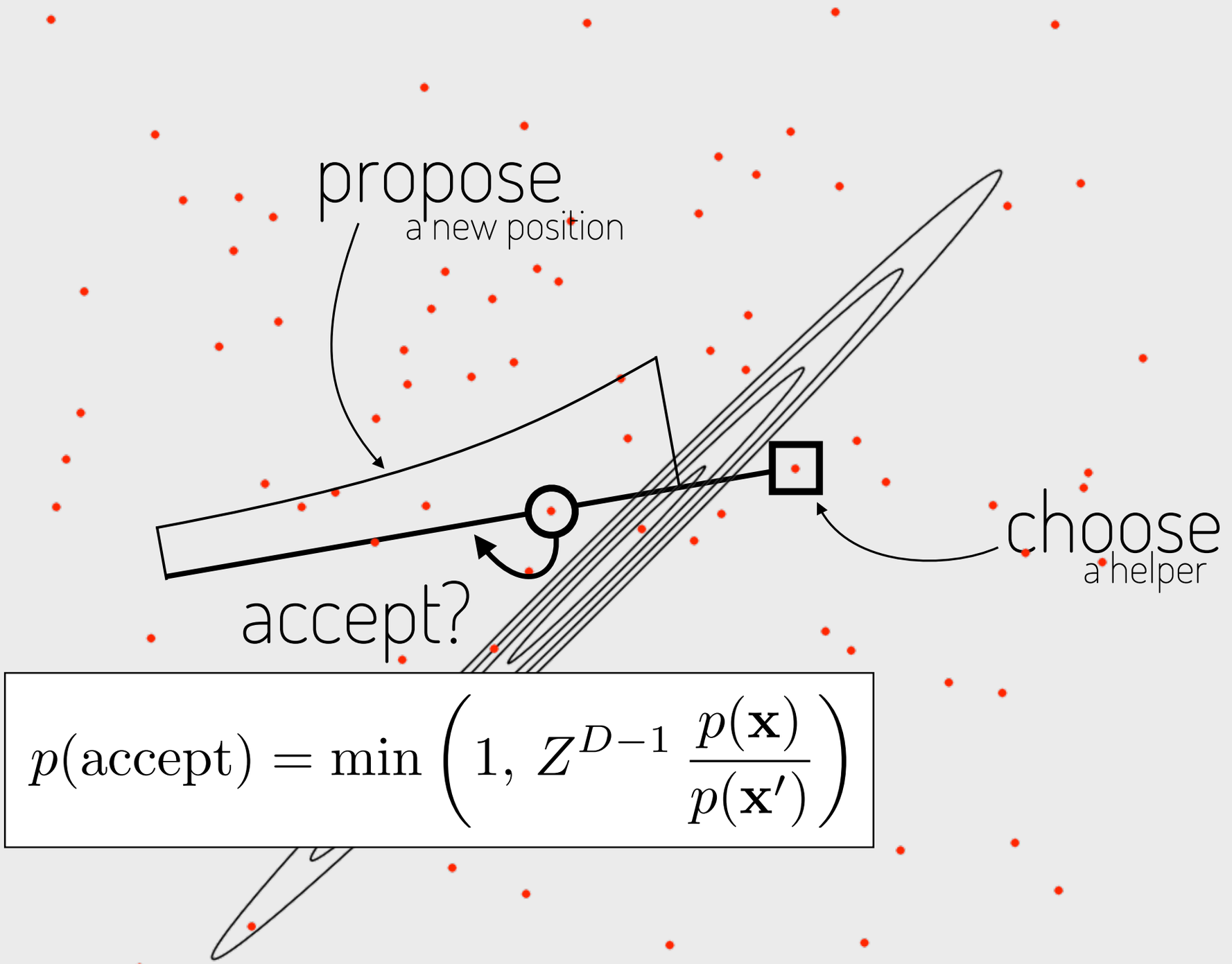
Ensemble Samplers

in the **real** world



Ensemble Samplers

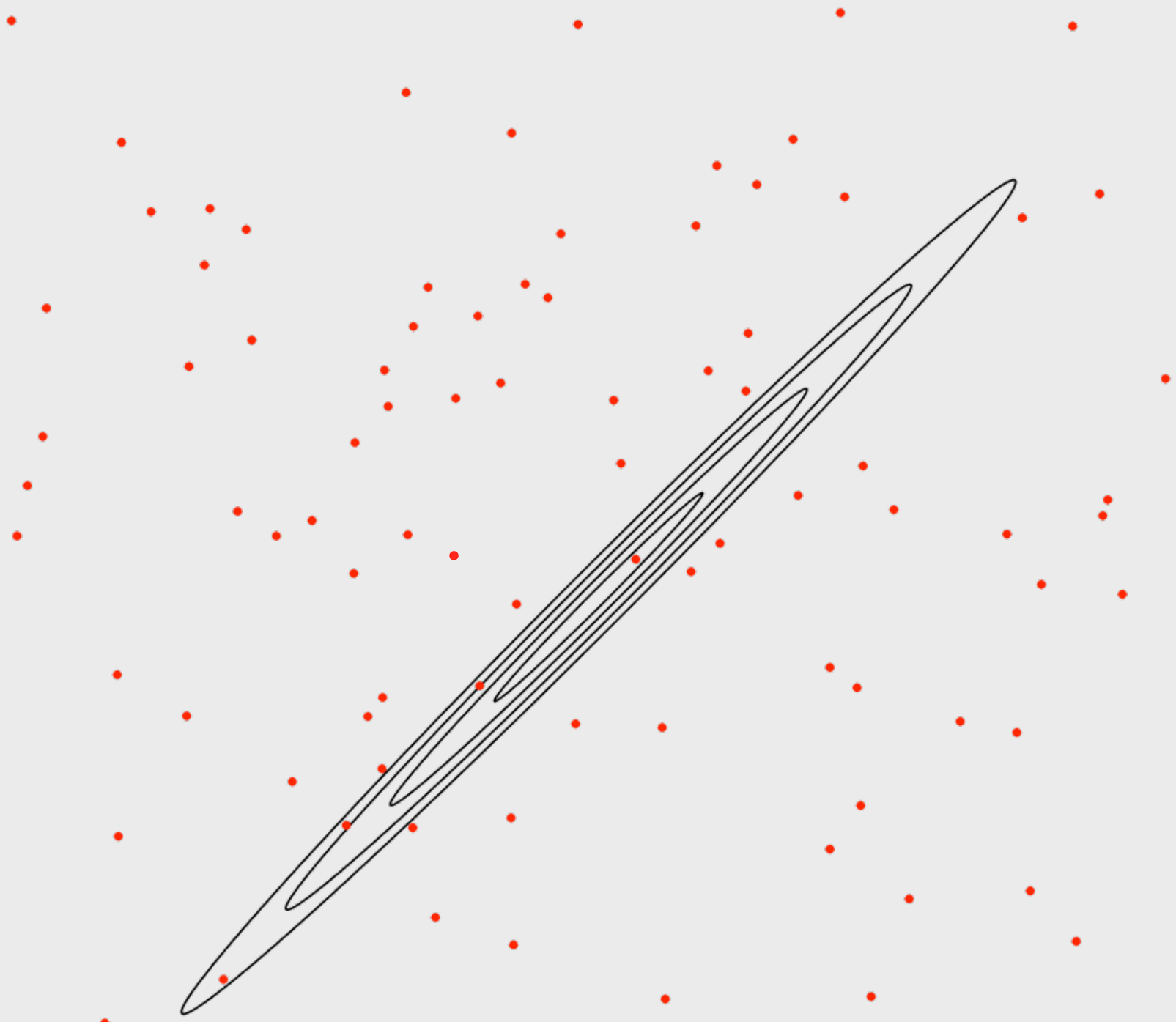
in the **real** world



$$p(\text{accept}) = \min\left(1,\, Z^{D-1}\,\frac{p(\mathbf{x})}{p(\mathbf{x}')}\right)$$

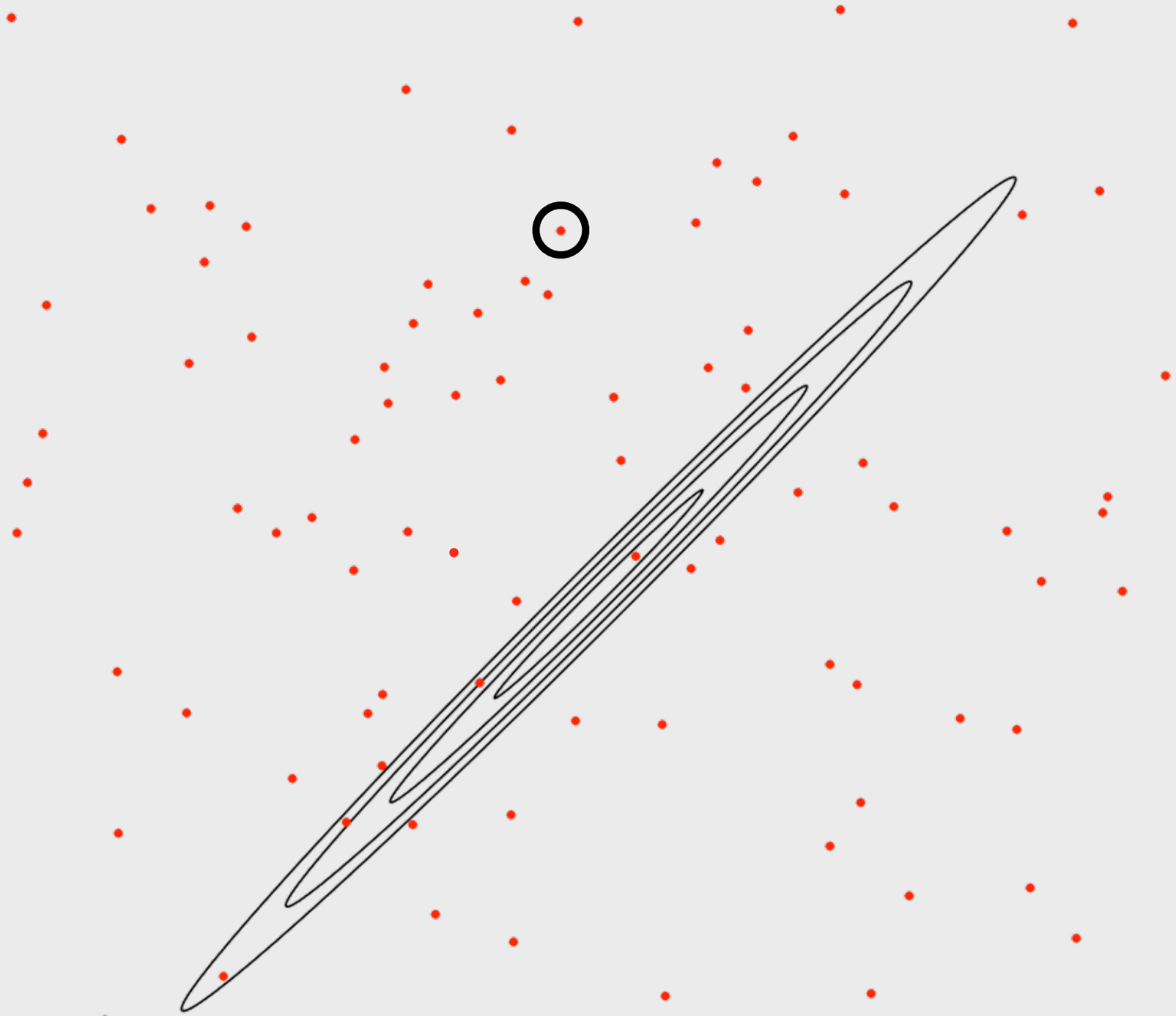
Ensemble Samplers

in the **real** world



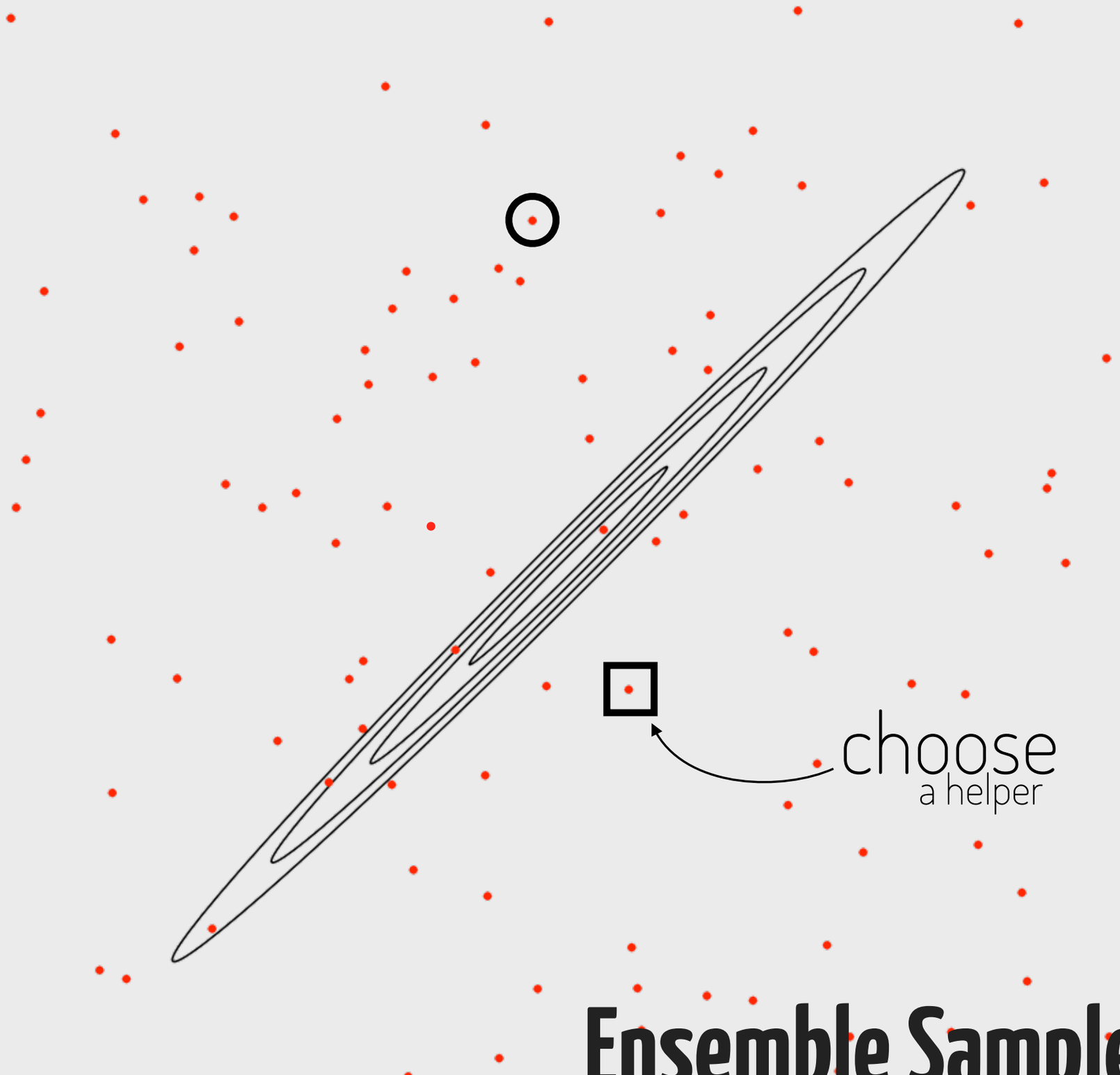
Ensemble Samplers

in the **real** world



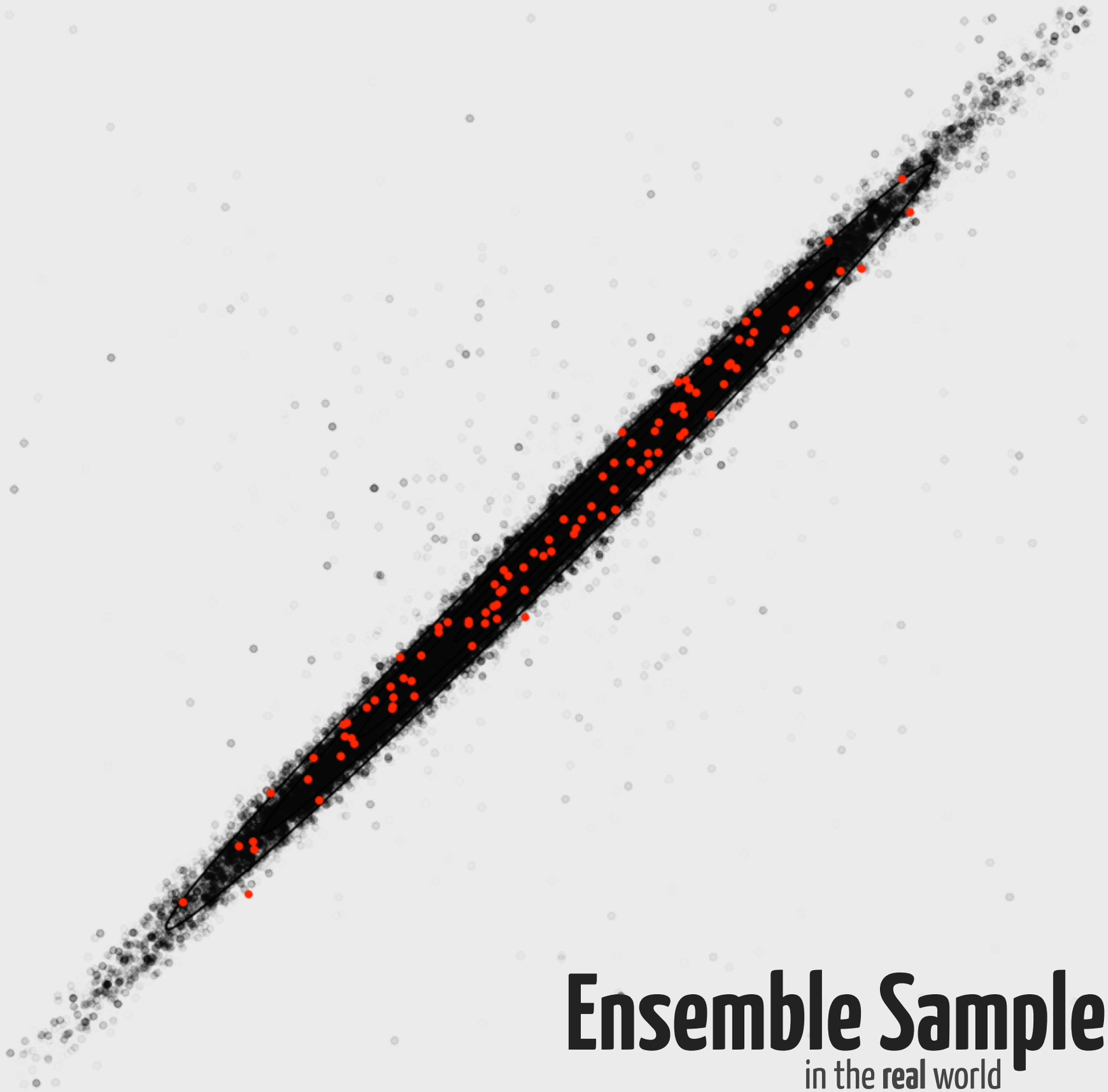
Ensemble Samplers

in the **real** world



Ensemble Samplers

in the **real** world



Ensemble Samplers

in the **real** world

go code up your own

Ensemble Sampler

code right now?

introducing **emcee** the MCMC Hammer

arxiv.org/abs/1202.3665

dan.iel.fm/emcee

to install:

```
pip install emcee
```

introducing **emcee** the MCMC Hammer

arxiv.org/abs/1202.3665

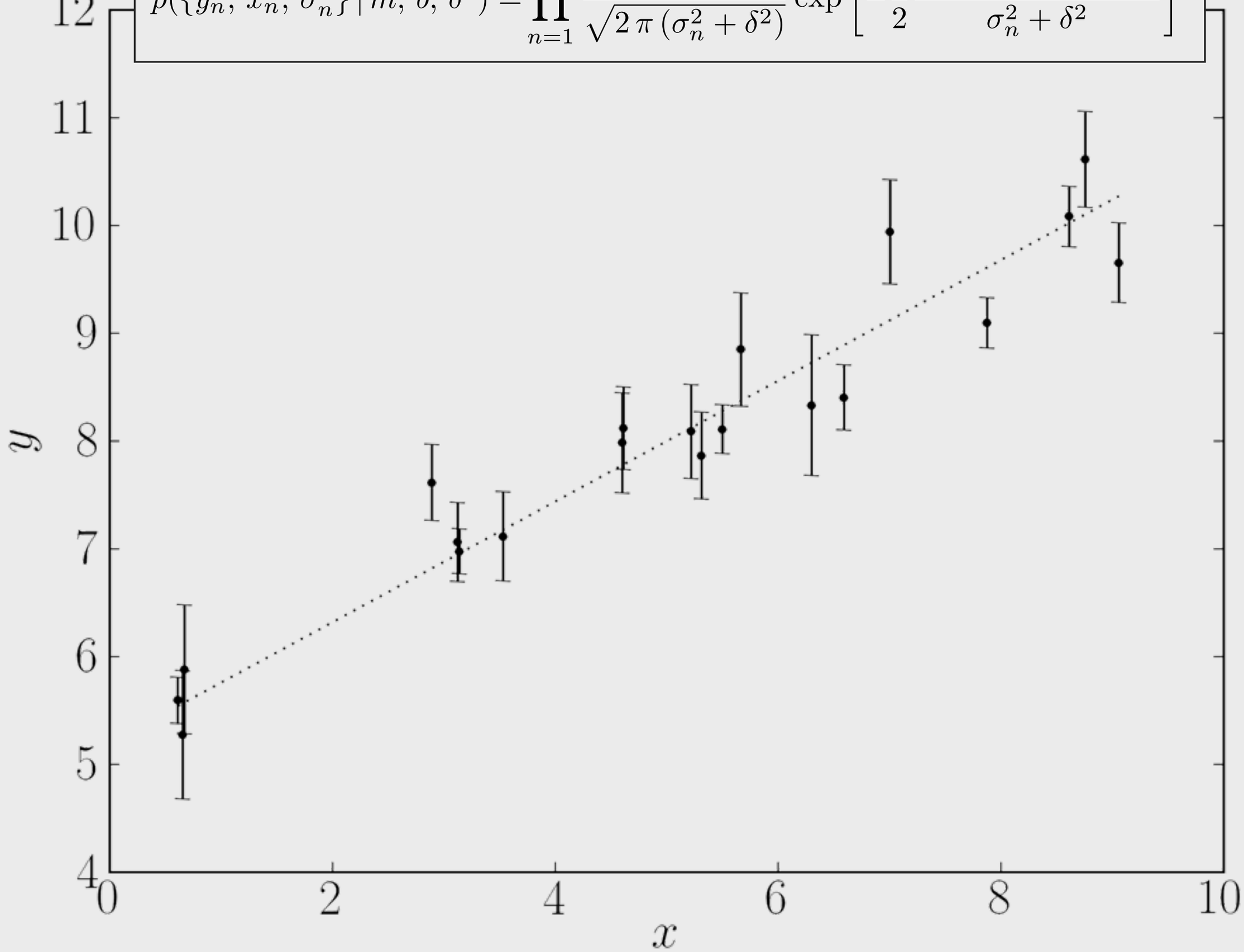
dan.iel.fm/emcee

using **emcee**

is **easy**

let me show you...

$$p(\{y_n,\, x_n,\, \sigma_n^2\}\,|\,m,\, b,\, \delta^2) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\,\pi\,(\sigma_n^2 + \delta^2)}} \exp\left[-\frac{1}{2}\,\frac{[y_n - (m\,x_n + b)]^2}{\sigma_n^2 + \delta^2}\right]$$



```python
import numpy as np
import emcee

def lnprobfn(p, x, y, yerr):
    m, b, d2 = p

    if not 0 <= d2 <= 1:
        return -np.inf

    ivar = 1.0 / (yerr ** 2 + d2)
    chi2 = np.sum((y - m * x - b) ** 2 * ivar)

    return -0.5 * (chi2 - np.sum(np.log(ivar)))

# Load data.
# x, y, yerr = ...

# Set up sampler and initialize the walkers.
nwalkers, ndim = 100, 3
sampler = emcee.EnsembleSampler(nwalkers, ndim, lnprobfn, args=(x, y, yerr))
p0 = [[np.random.rand(), 10 * np.random.rand(), np.random.rand()]
      for k in range(nwalkers)]

# Go.
sampler.run_mcmc(p0, 1000)
m, b, d2 = sampler.flatchain.T
```

$$\ln p(\{x_n,\, y_n,\, \sigma_n\}\,|\,m,\, b,\, \delta^2) = -\frac{1}{2}\chi^2 - \frac{1}{2}\sum_{n=1}^{N} \ln\left(\sigma_n^2 + \delta^2\right)$$

$$\chi^2 = \sum_{n=1}^{N} \frac{(y_n - m\,x_n + b)^2}{\sigma_n^2 + \delta^2}$$

$$p(\delta^2) = \begin{cases} 1, & \text{if } 0 \le \delta^2 \le 1 \\ 0, & \text{otherwise} \end{cases}$$

```python
import numpy as np

def lnprobfn(p, x, y, yerr):
    m, b, d2 = p

    if not 0 <= d2 <= 1:
        return -np.inf

    ivar = 1.0 / (yerr ** 2 + d2)
    chi2 = np.sum((y - m * x - b) ** 2 * ivar)

    return -0.5 * (chi2 - np.sum(np.log(ivar)))
```

```python
import emcee

nwalkers, ndim = 100, 3
sampler = emcee.EnsembleSampler(nwalkers, ndim, lnprobfn,
                                args=(x, y, yerr))

p0 = [[np.random.rand(), 10 * np.random.rand(), np.random.rand()]
      for k in range(nwalkers)]

sampler.run_mcmc(p0, 1000)
m, b, d2 = sampler.flatchain.T
```

initialize the sampler

initialize each walker

$$m_k \sim \mathcal{U}(0, 1)$$

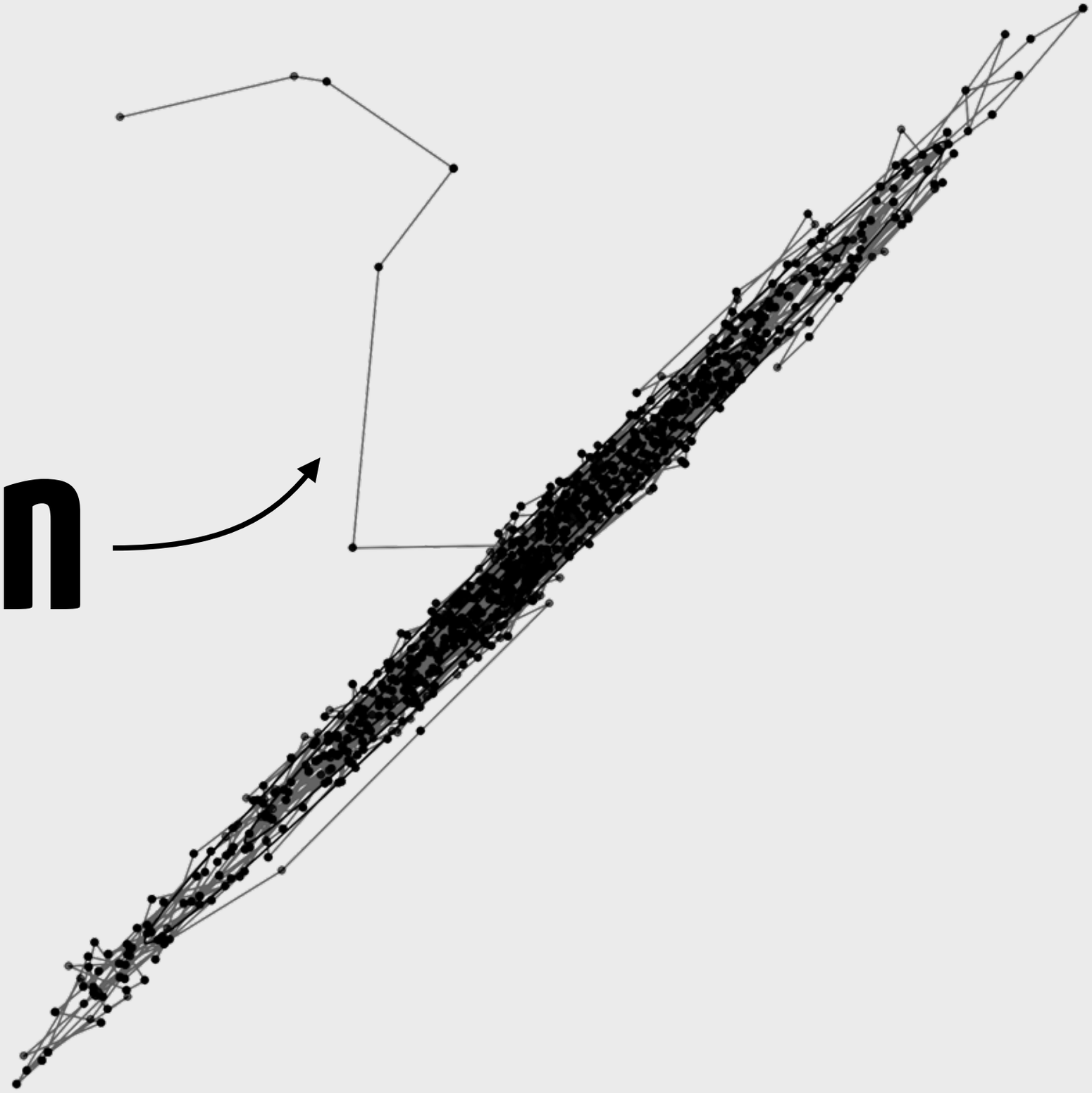
$$b_k \sim \mathcal{U}(0, 10)$$

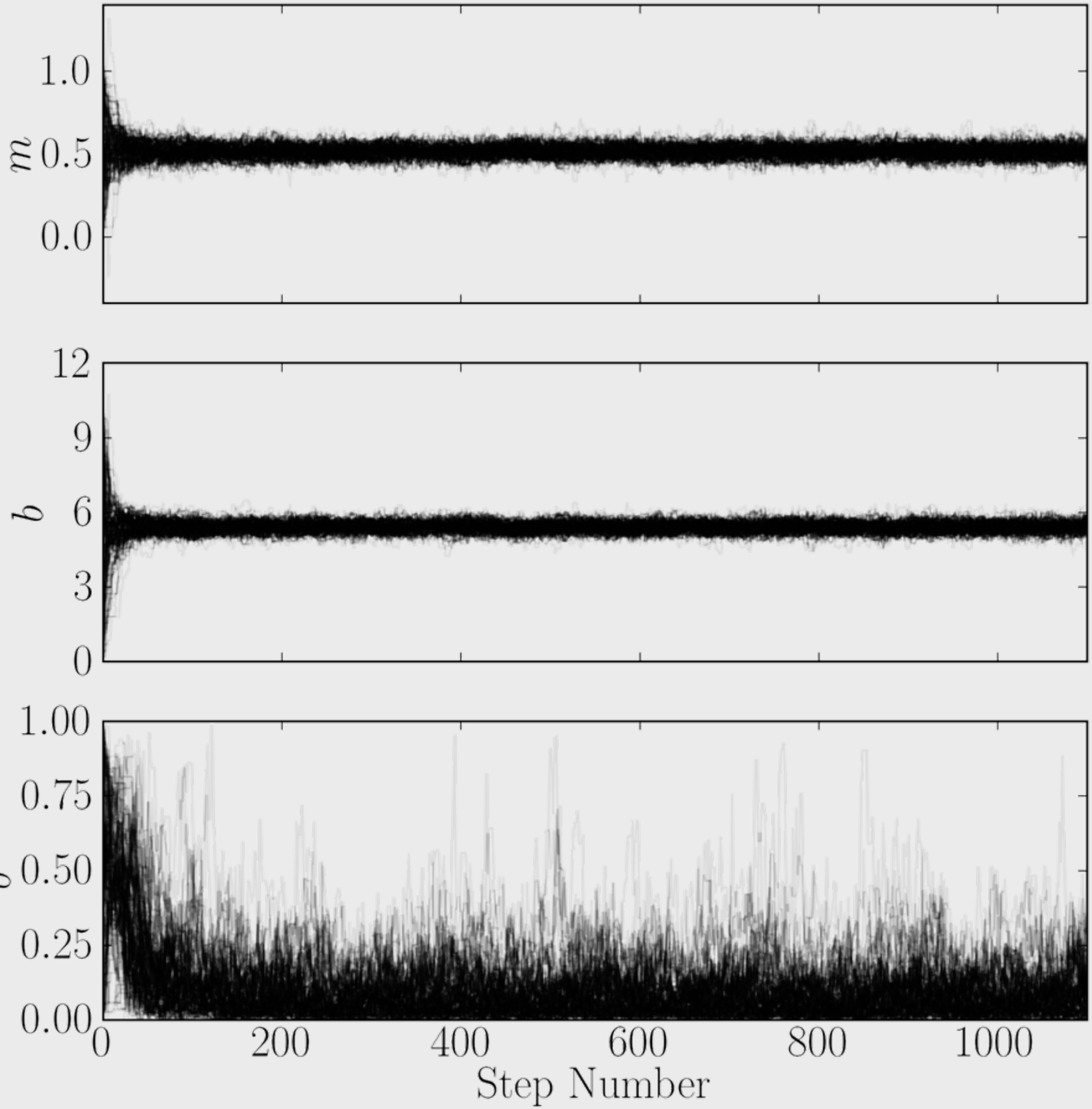
$$\delta_k^2 \sim \mathcal{U}(0, 1)$$

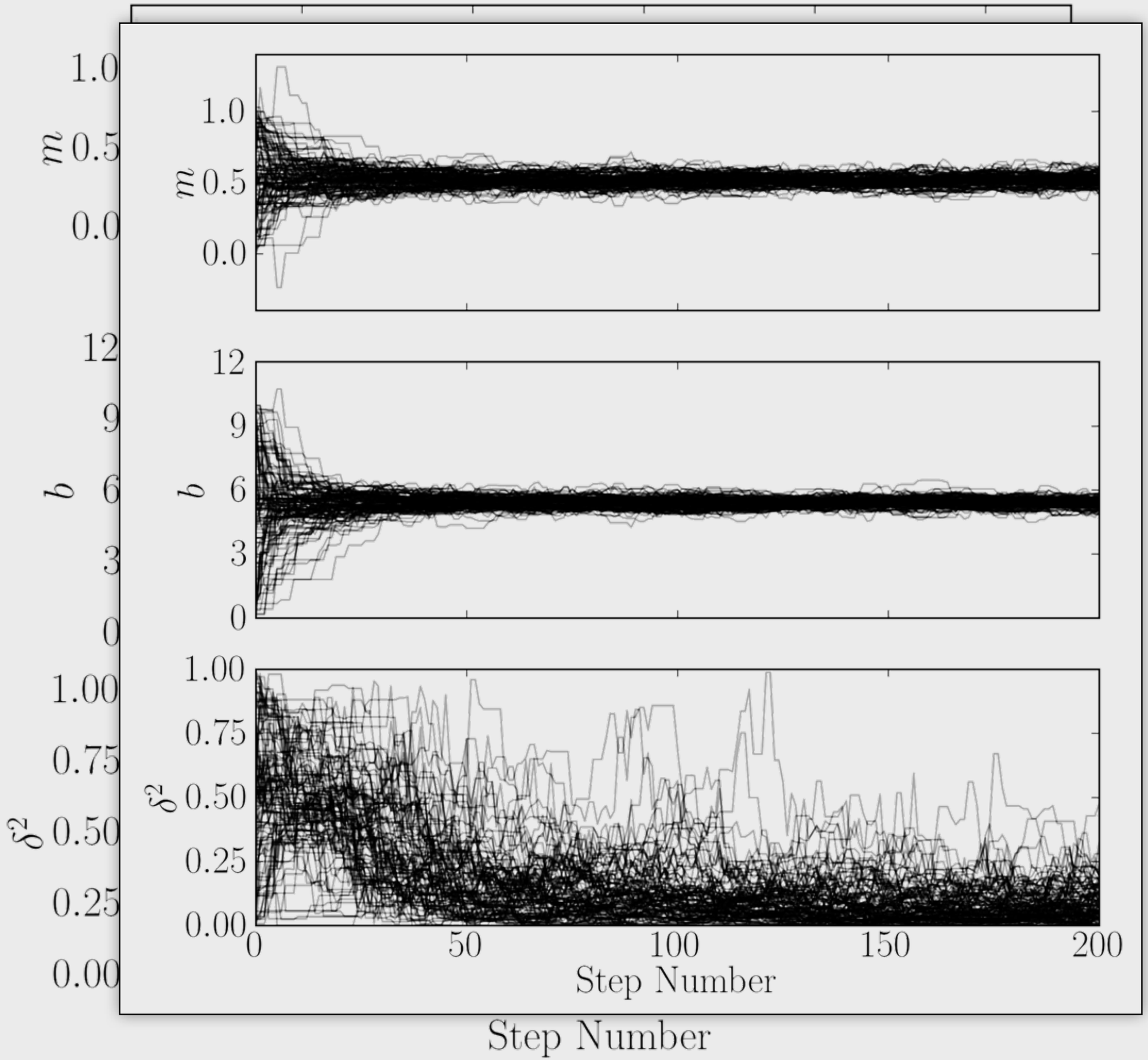
run MCMC for 1000 steps

get the samples

burn-in




m
1.0
0.5
0.0
b
12
9
6
3
0
δ^2
1.00
0.75
0.50
0.25
0.00
0
200
400
600
800
1000
Step Number


m
1.0
0.5
0.0
b
12
9
6
3
0
δ^2
1.00
0.75
0.50
0.25
0.00
0
50
100
150
200
Step Number

convergence

acceptance fraction?

The limiting diffusion approximation admits a straight-forward efficiency maximization problem, and the resulting asymptotically optimal policy is related to the asymptotic acceptance rate of proposed moves for the algorithm. The asymptotically optimal acceptance rate is 0.234 under quite general conditions.

acceptance fraction?

The limiting diffusion approximation admits a straight-forward efficiency maximization problem, and the resulting asymptotically optimal policy is related to the asymptotic acceptance rate of proposed moves for the algorithm. The asymptotically optimal acceptance rate is 0.234 under quite general conditions.

whoa!

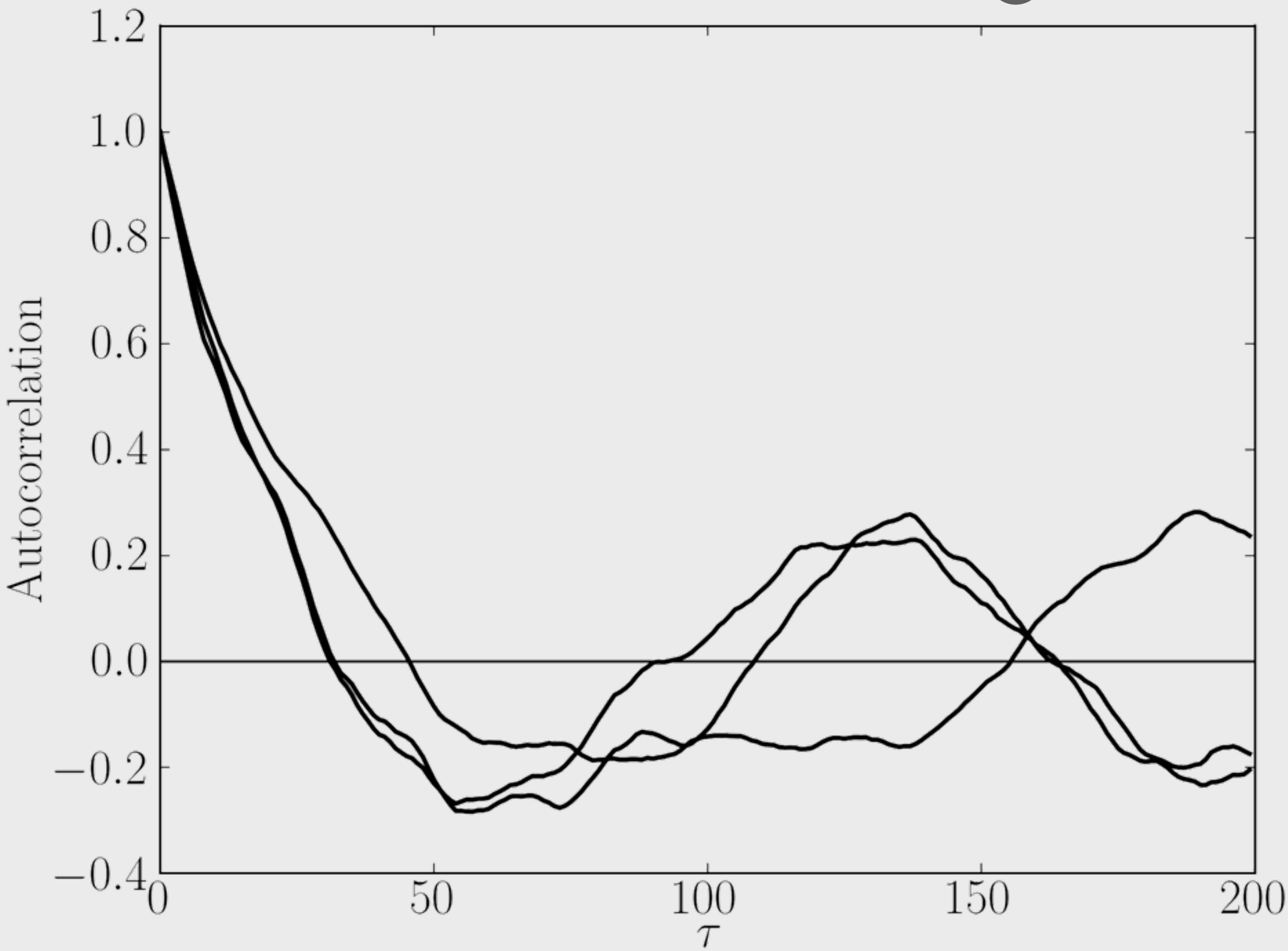
acceptance fraction?

The limiting diffusion approximation admits a straight-forward efficiency maximization problem, and the resulting asymptotically optimal policy is related to the asymptotic acceptance rate of proposed moves for the algorithm. The asymptotically optimal acceptance rate is 0.234 under quite general conditions.

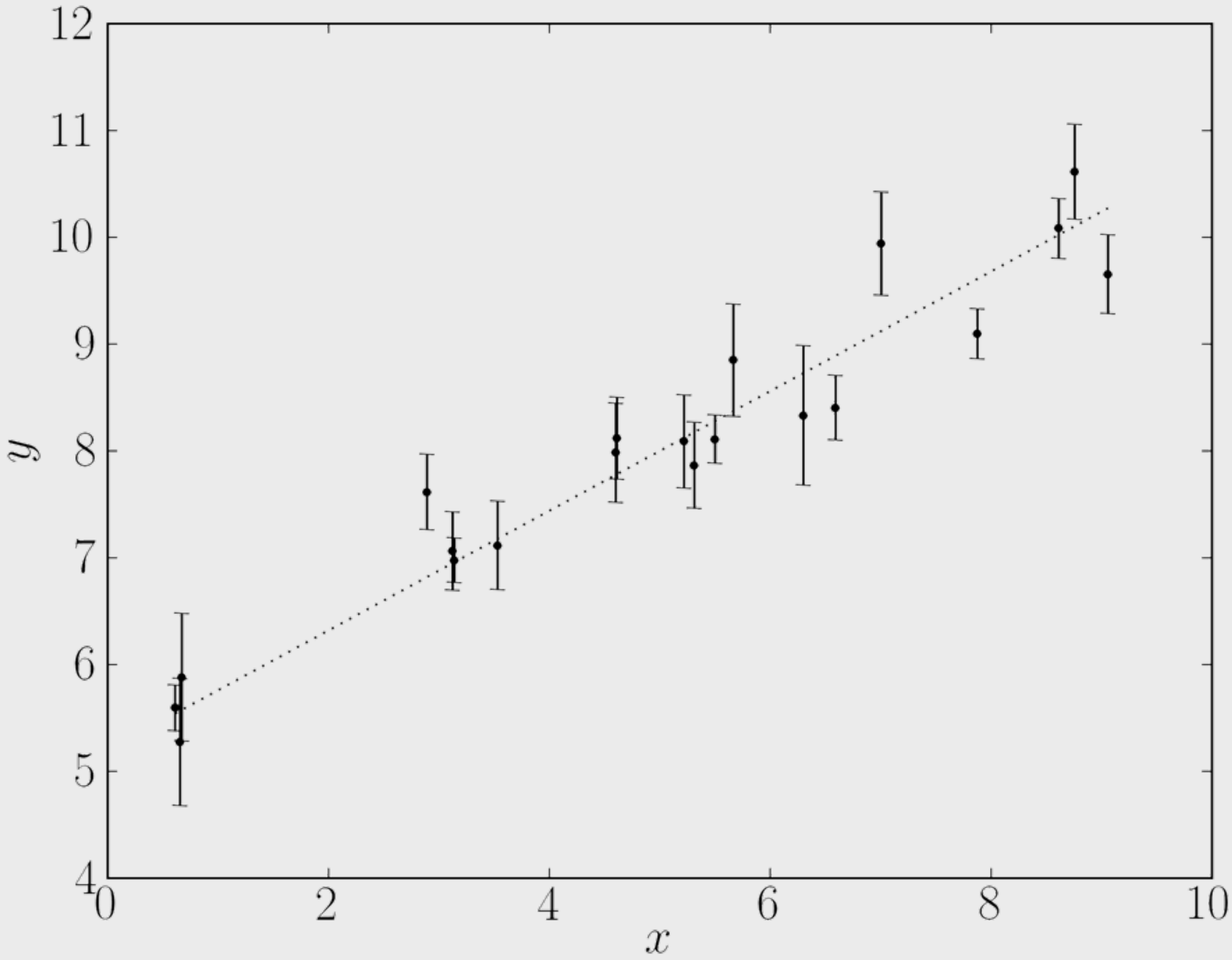
whoa!

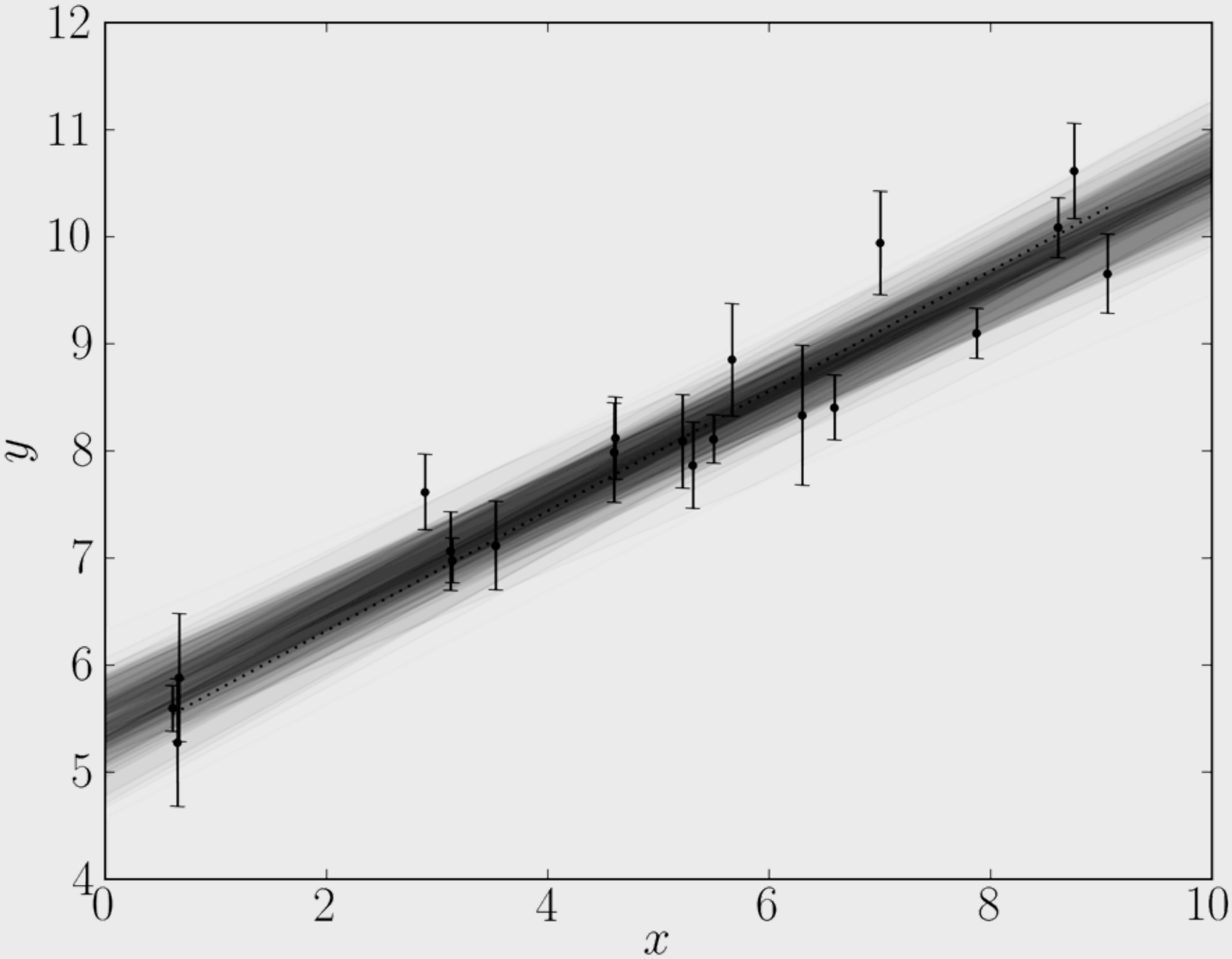
that's a lot of significant digits...

dfm/acor

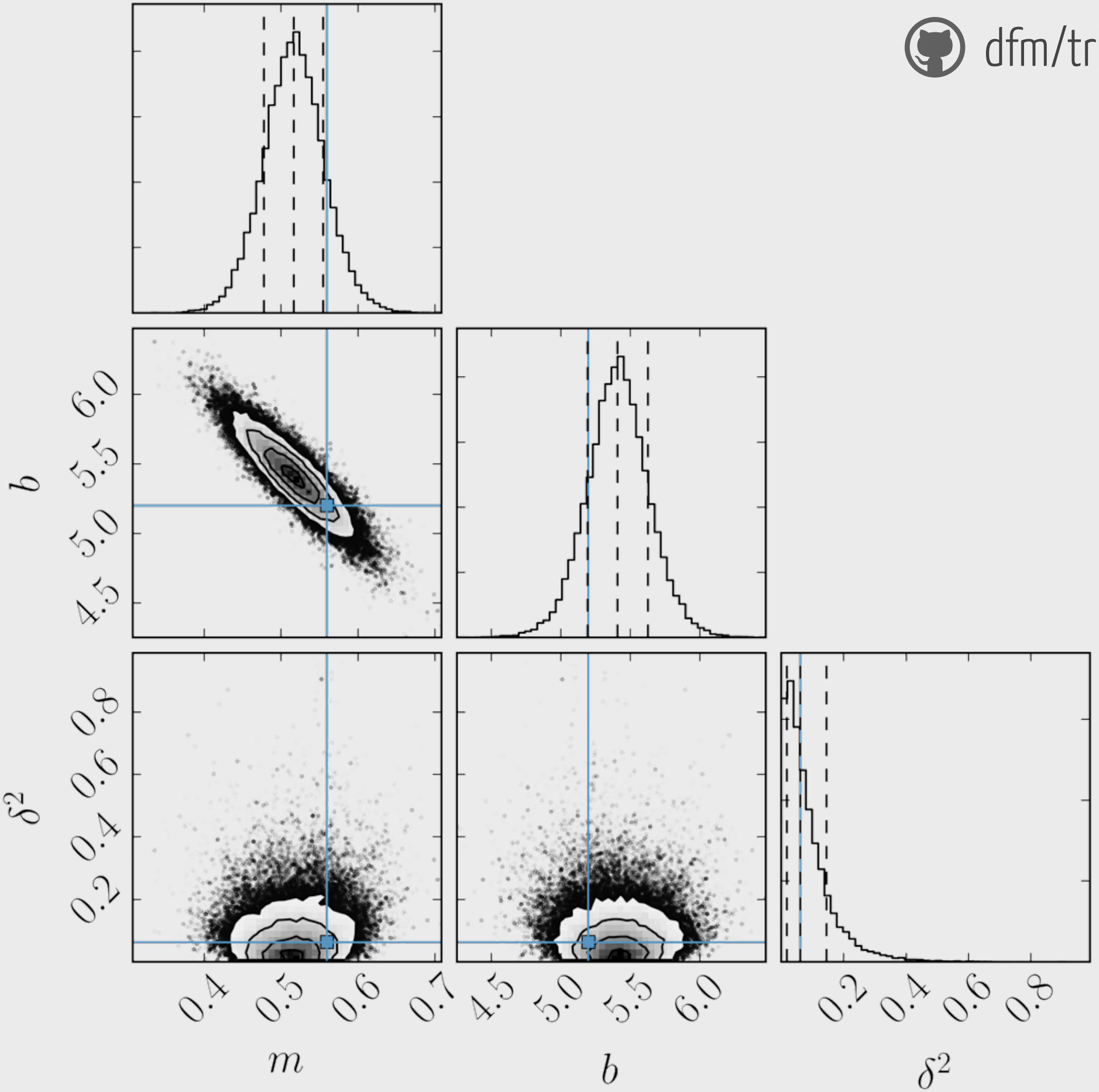


displaying results


12
11
10
9
8
7
6
5
4
y
0
2
4
6
8
10
x


12
11
10
9
8
7
6
5
4
y
0
2
4
6
8
10
x

dfm/triangle.py



sharing results

you need a number for your abstract?!?

1. sort the samples
2. compute moments/quantiles

(0.16, 0.5, 084)

1. sort the samples
2. compute moments/quantiles **(0.16, 0.5, 084)**

$$X = \bar{X}^{+\delta_+}_{-\delta_-}$$

what does it MEAN?

1. sort the samples
2. compute moments/quantiles
   (0.16, 0.5, 084)

$$X = \bar{X}^{+\delta_+}_{-\delta_-}$$

what does it MEAN?

3. machine readable samplings
   (including priors values)

1. sort the samples
2. compute moments/quantiles
   (0.16, 0.5, 084)

$$X = \bar{X}^{+\delta_+}_{-\delta_-}$$

what does it MEAN?

3. machine readable samplings
   (including priors values)

my pipe dream

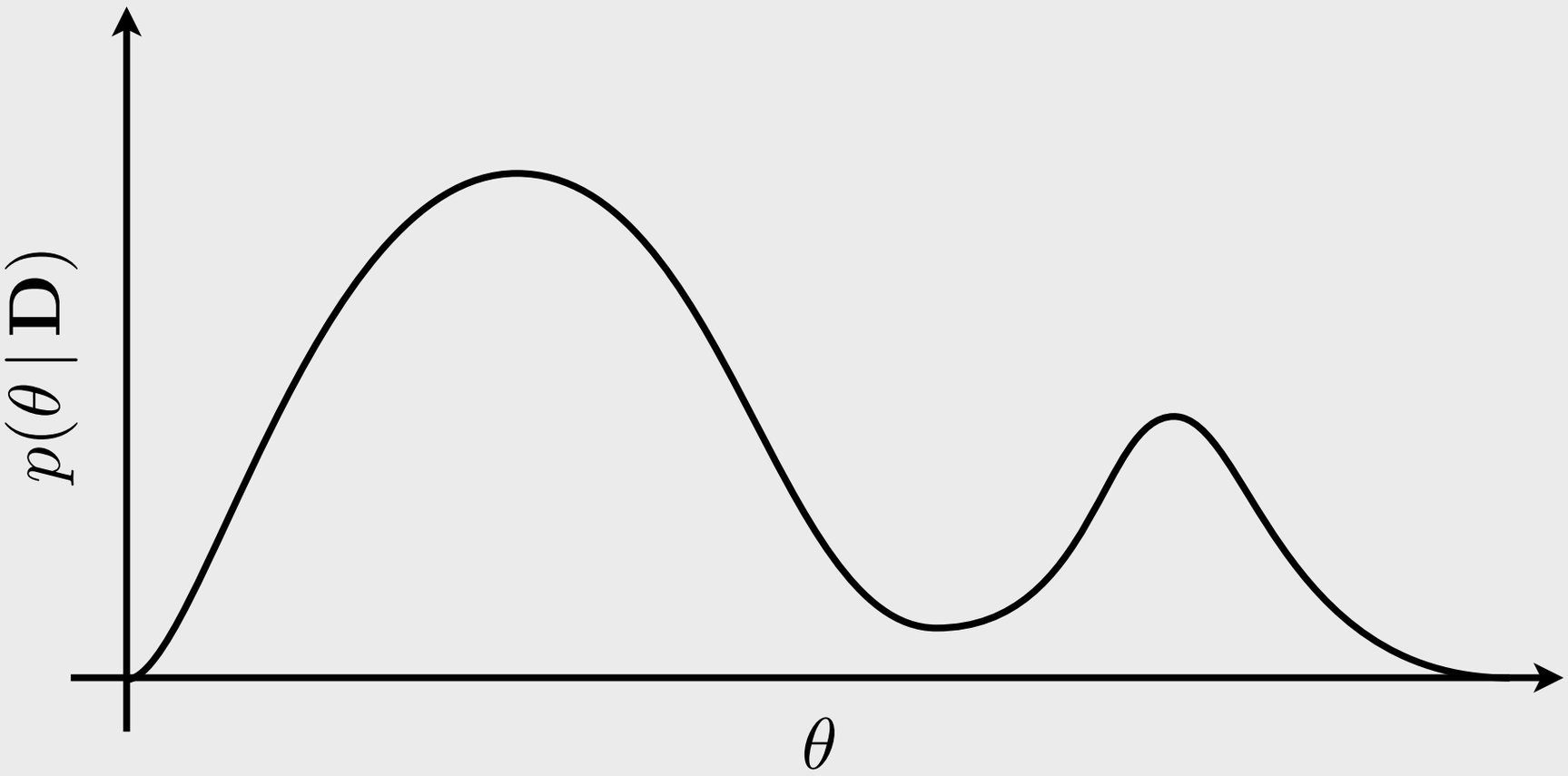
Brendon Brewer's words of wisdom...

Remember:

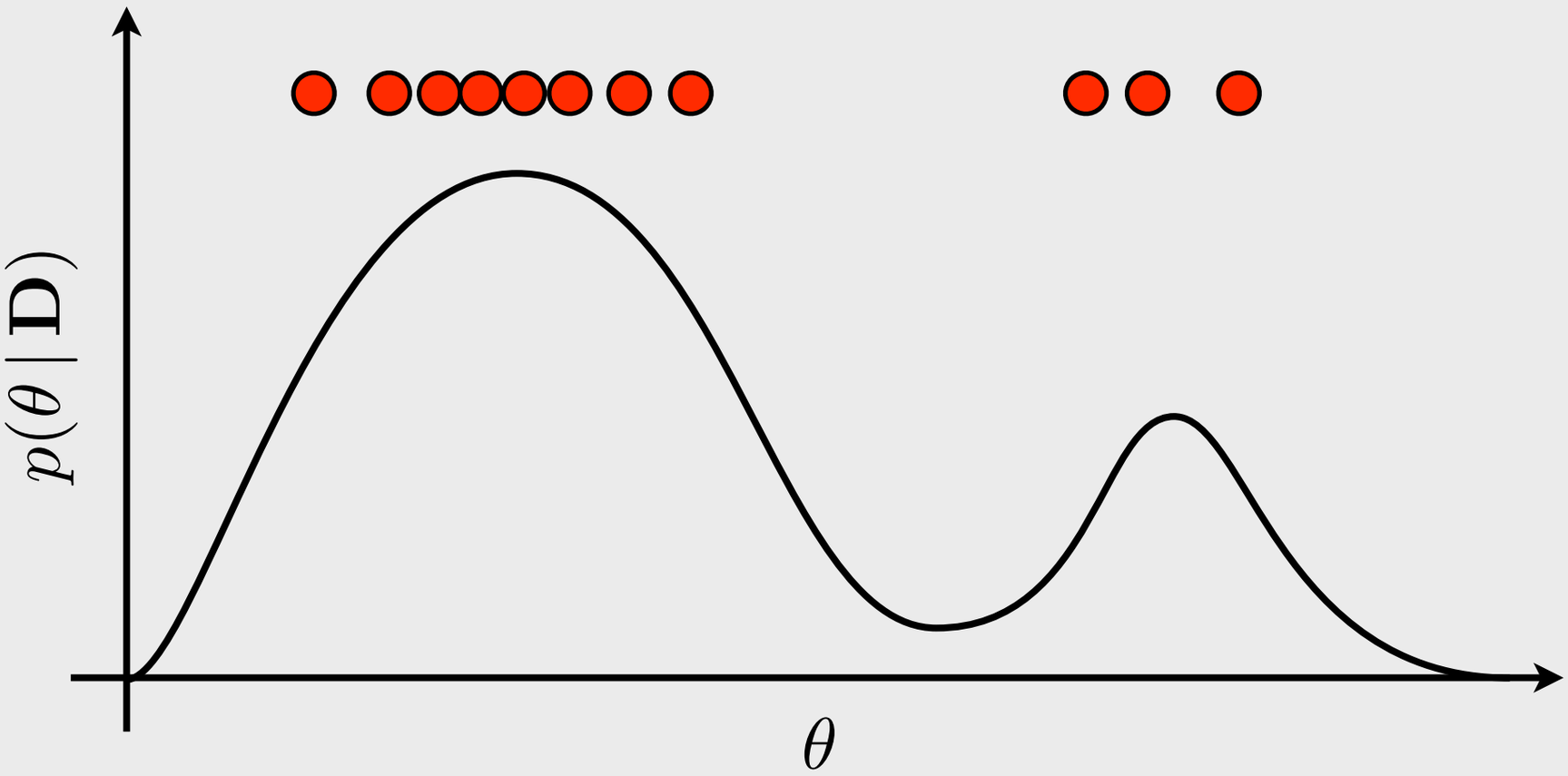
emcee isn't always The Right Choice™

Brendon Brewer's words of wisdom...

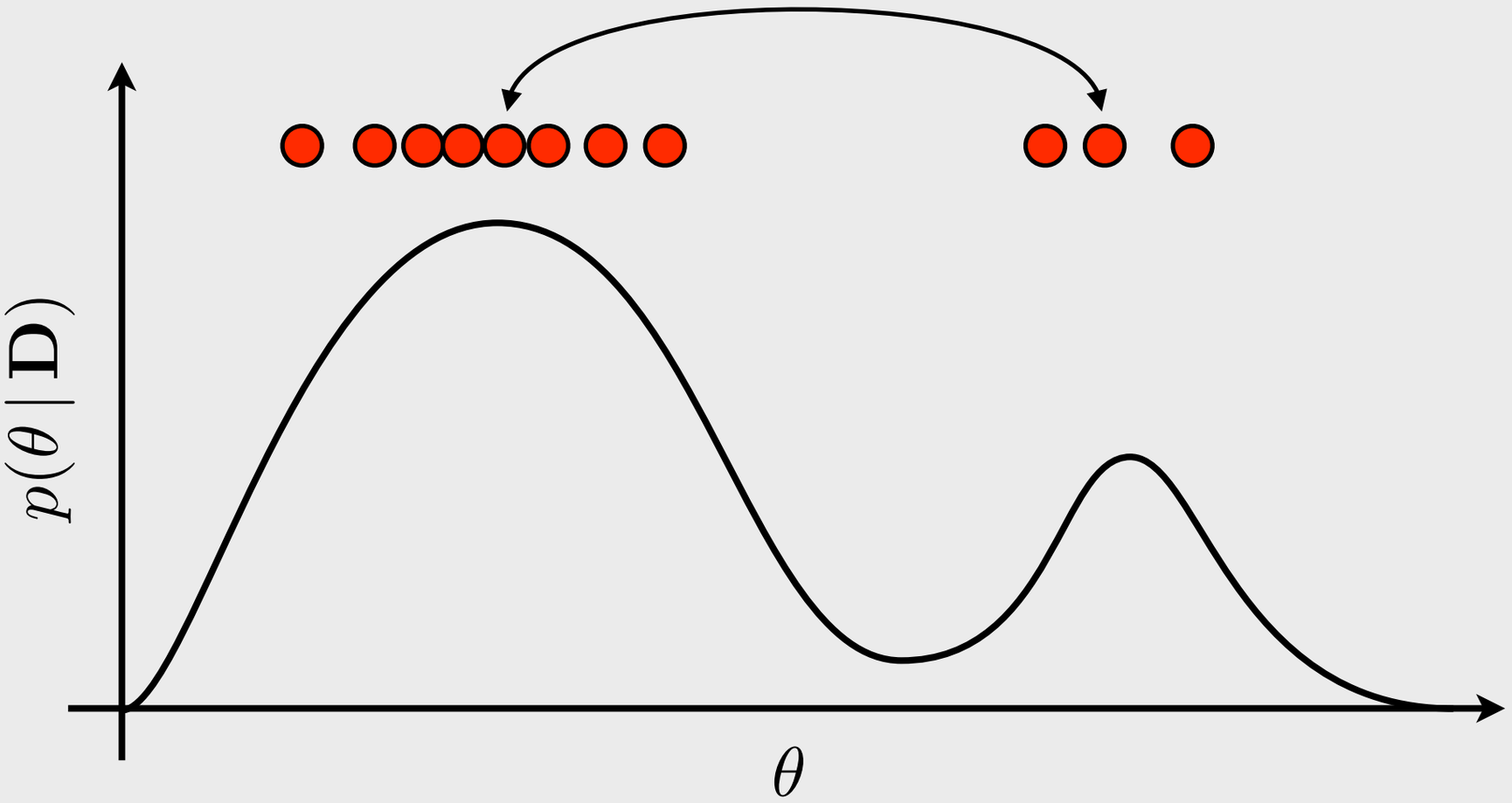
what about multimodal densities?



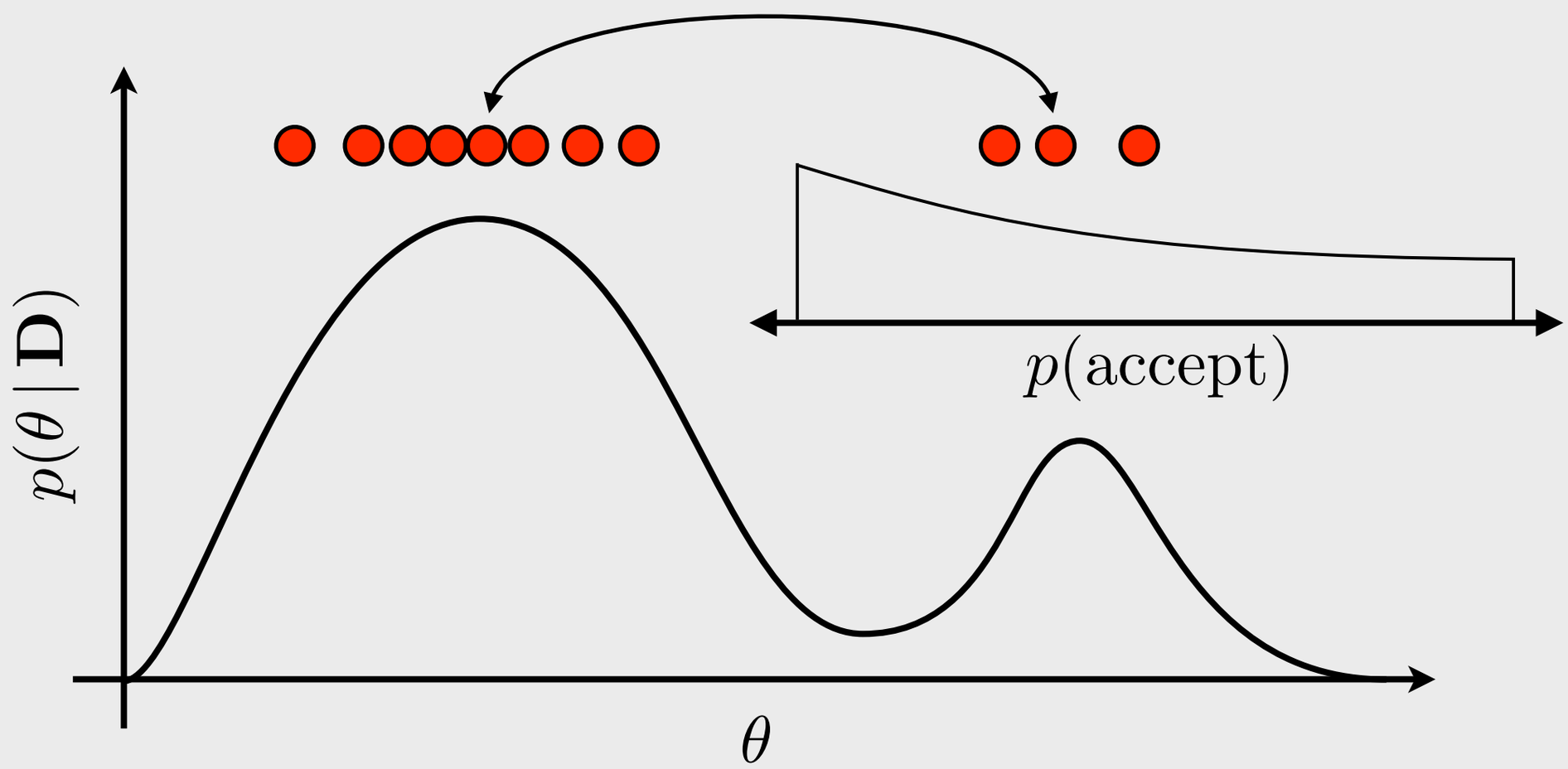
what about multimodal densities?



what about multimodal densities?



what about multimodal densities?



what about multimodal densities?

what if we want to compare models?

$$p(\boldsymbol{\theta} \,|\, \mathbf{D},\, M) = \frac{1}{Z(M)}\, p(\boldsymbol{\theta} \,|\, M)\, p(\mathbf{D} \,|\, \boldsymbol{\theta},\, M)$$

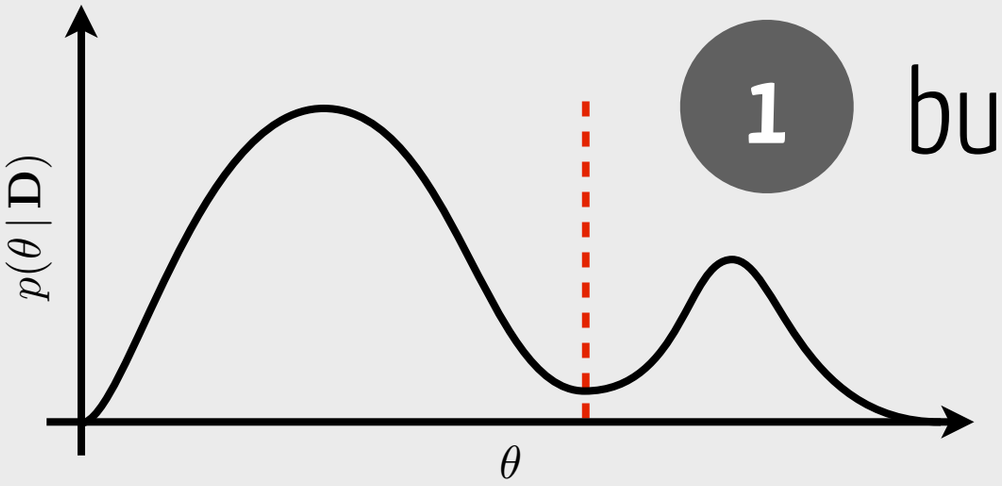
$$Z = p(\mathbf{D} \,|\, M) = \int p(\mathbf{D},\, \boldsymbol{\theta} \,|\, M)\, \mathrm{d}\boldsymbol{\theta}$$

what if we want to compare models?

$$p(\boldsymbol{\theta} \mid \mathbf{D}, \qquad \mid \boldsymbol{\theta},\ M)$$

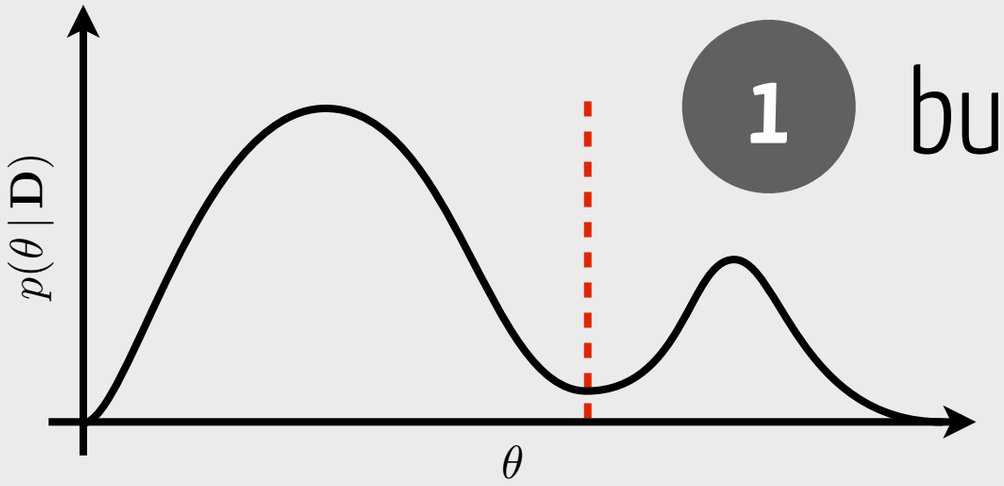
$$Z = \qquad \Big) \, \mathrm{d}\boldsymbol{\theta}$$

what to do?



1 burn-in & priors

what to do?



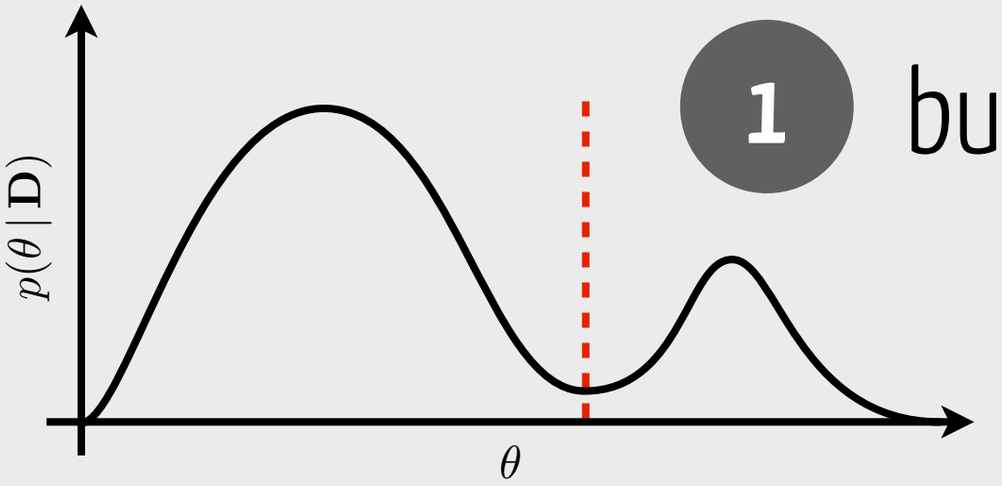
1 burn-in & priors

what to do?

use a **different algorithm** 2

(gasp!)

$$Z = p(\mathbf{D} \mid M) = \int p(\mathbf{D}, \boldsymbol{\theta} \mid M)\, \mathrm{d}\boldsymbol{\theta}$$



1 burn-in & priors

what to do?

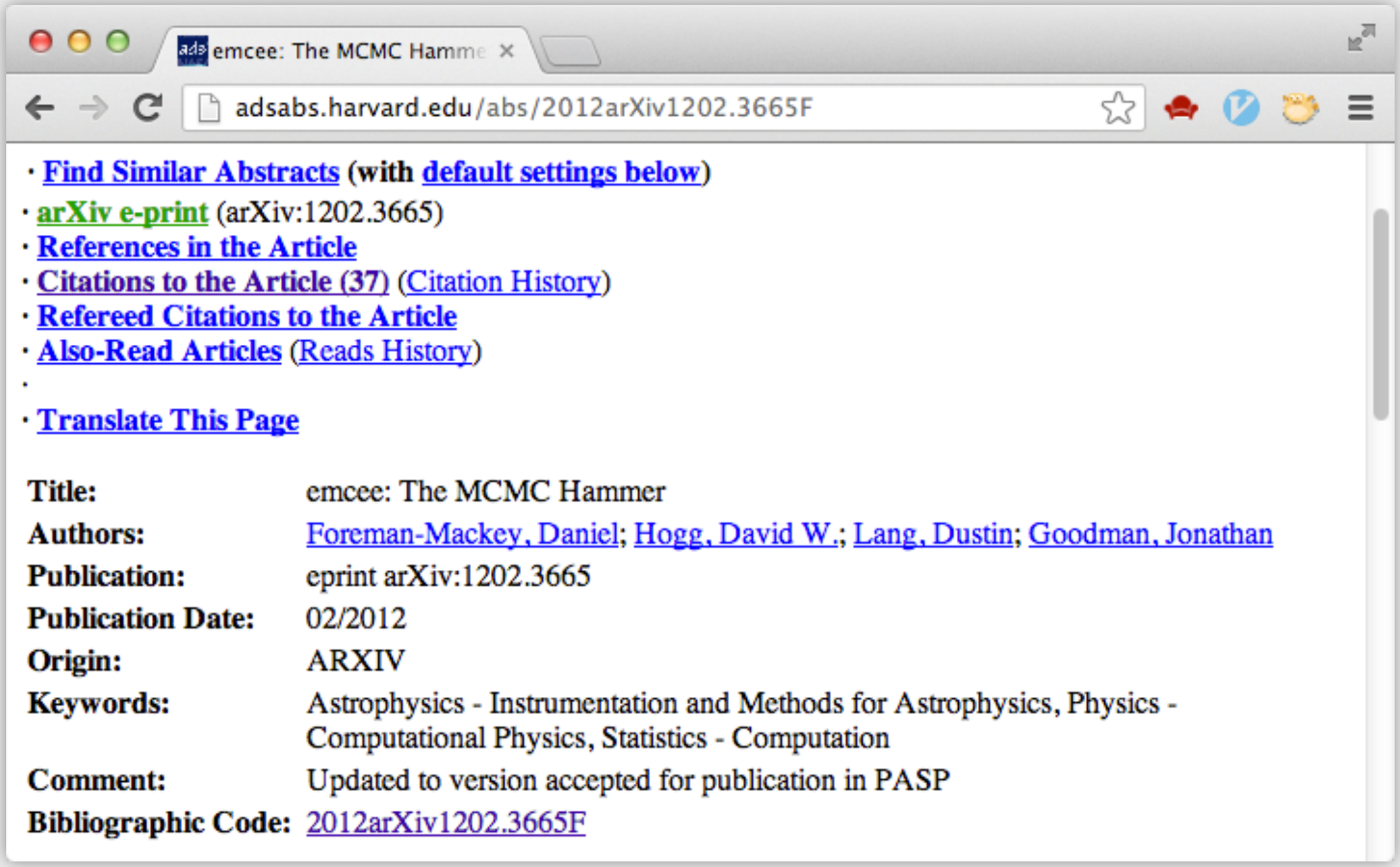
use a **different algorithm** 2

(gasp!)

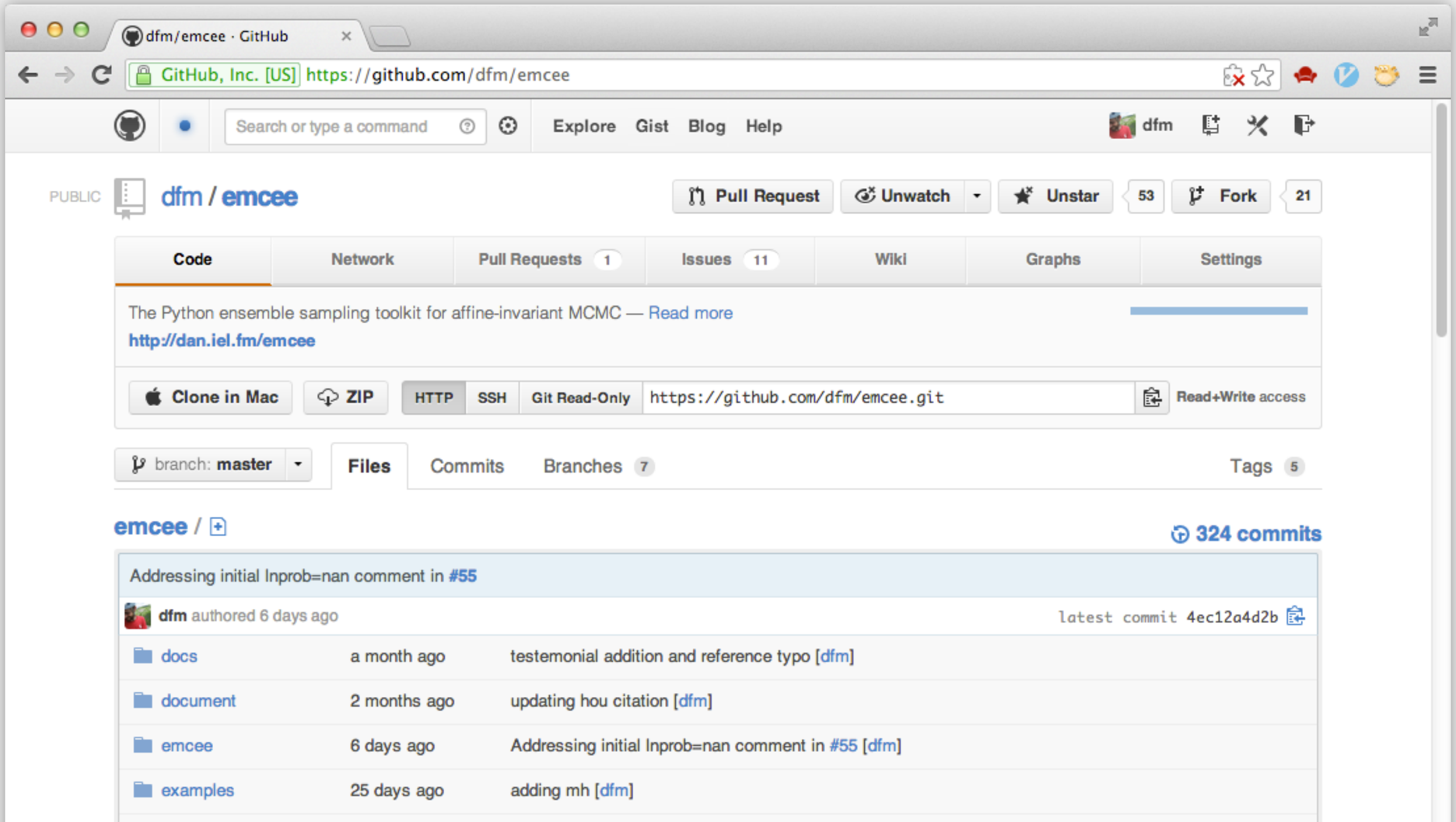
$$Z = p(\mathbf{D} \mid M) = \int p(\mathbf{D}, \boldsymbol{\theta} \mid M)\, \mathrm{d}\boldsymbol{\theta}$$

github.com/eggplantbren/DNest3

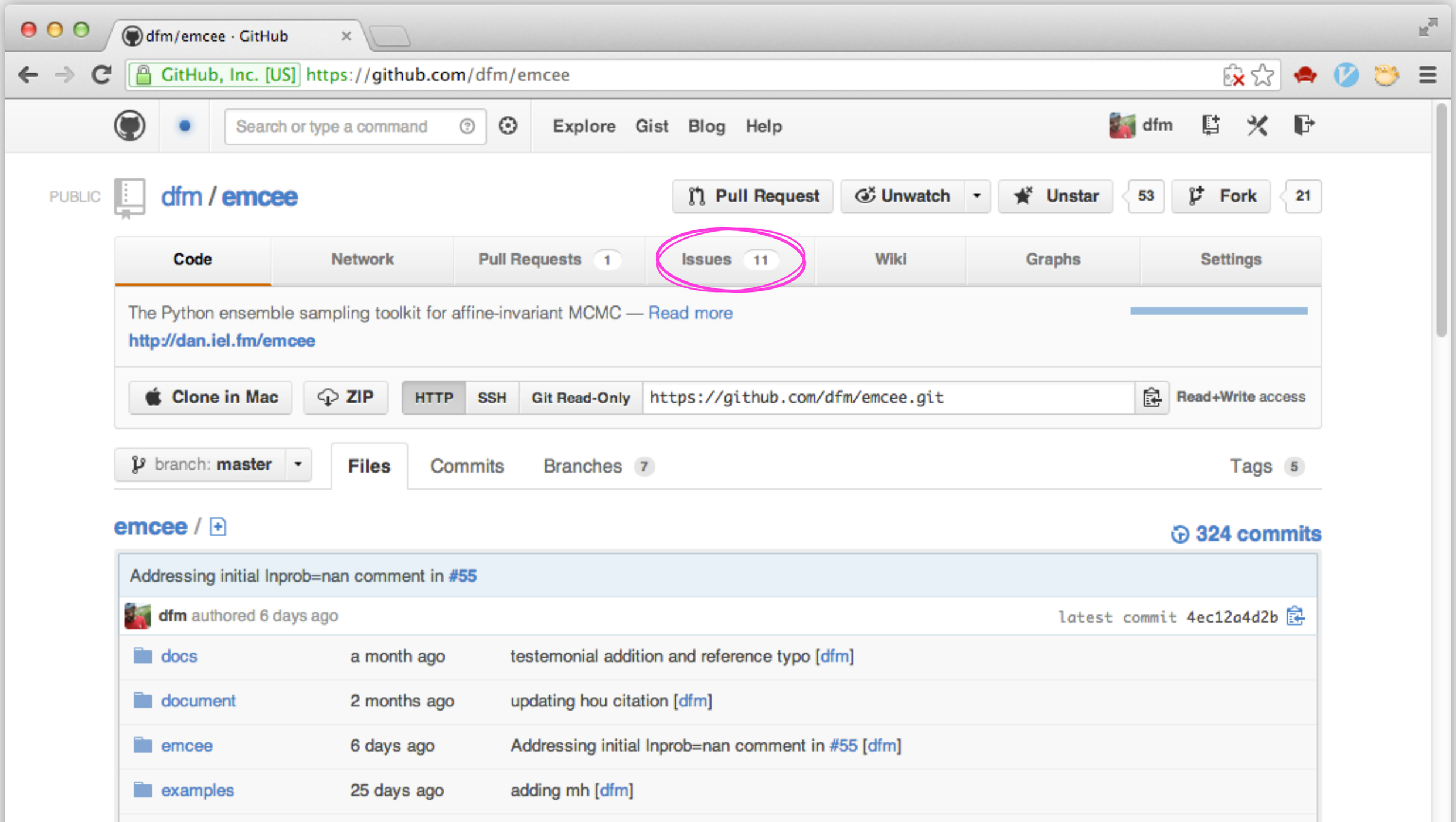
sometimes it is useful



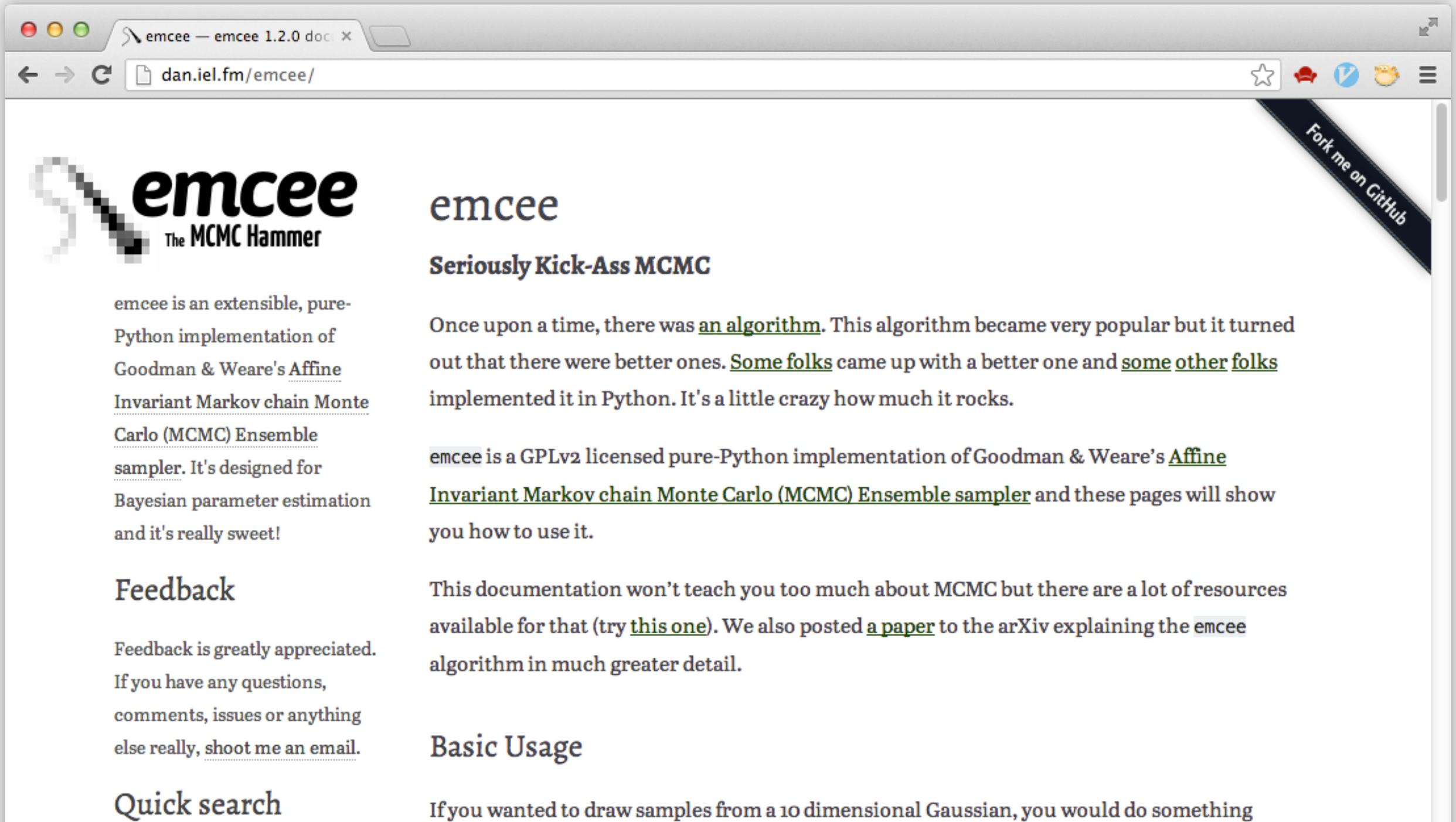
emcee is community supported.



emcee is community supported.



emcee has good documentation.



emcee has a live support team.

danfm@nyu.edu

Alex Conley (UC Boulder)
Jason Davies (Jason Davies Ltd.)
Will Meierjurgen Farr (Northwestern)
David W. Hogg (NYU)
Dustin Lang (CMU)
Phil Marshall (Oxford)
Ilya Pashchenko (ASC LPI, Moscow)
Adrian Price-Whelan (Columbia)
Jeremy Sanders (Cambridge)
Joe Zuntz (Oxford)

Eric Agol (UW)
Jo Bovy (IAS)
Jacqueline Chen (MIT)
John Gizis (Delaware)
Jonathan Goodman (NYU)
Marius Millea (UC Davis)
Jennifer Piscionere (Vanderbilt)

contributors

take home.

your model:

$$p(\mathbf{D} \mid \boldsymbol{\theta}, \boldsymbol{\alpha})$$

take home.

your model:

$$p(\mathbf{D} \mid \boldsymbol{\theta}, \boldsymbol{\alpha})$$

$$p(\mathbf{D}|\boldsymbol{\theta}) \propto \int p(\boldsymbol{\alpha})\, p(\mathbf{D}|\boldsymbol{\theta}, \boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\alpha}$$

take home.

your model:

$$p(\mathbf{D} \mid \boldsymbol{\theta}, \boldsymbol{\alpha})$$

$$p(\mathbf{D}|\boldsymbol{\theta}) \propto \int p(\boldsymbol{\alpha})\, p(\mathbf{D}|\boldsymbol{\theta}, \boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\alpha}$$

$$\boldsymbol{\theta} =$$

Alex Conley (UC Boulder)
Jason Davies (Jason Davies Ltd.)
Will Meierjurgen Farr (Northwestern)
David W. Hogg (NYU)
Dustin Lang (CMU)
Phil Marshall (Oxford)
Ilya Pashchenko (ASC LPI, Moscow)
Adrian Price-Whelan (Columbia)
Jeremy Sanders (Cambridge)
Joe Zuntz (Oxford)

Eric Agol (UW)
Jo Bovy (IAS)
Jacqueline Chen (MIT)
John Gizis (Delaware)
Jonathan Goodman (NYU)
Marius Millea (UC Davis)
Jennifer Piscionere (Vanderbilt)

thanks!