A Beginner's Guide to MCMC

David Kipping

Sagan Workshop 2016

but first, Sagan workshops, symposiums and fellowships are the bomb

how to get the most out of a Sagan workshop, 2009-style

lunch with Saganites

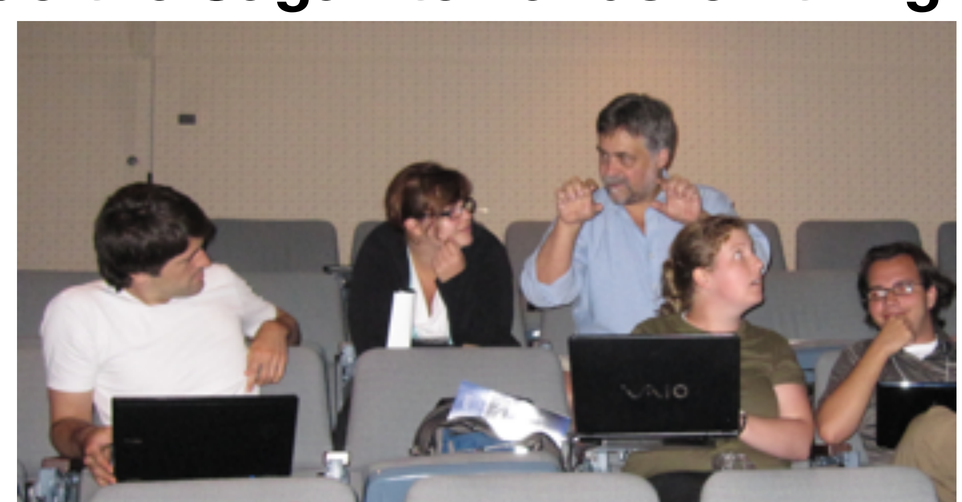
POP the Saganites

listen to the Saganites

drink coffee with Saganites

(perhaps bring more than one t-shirt for the whole week)

do the Saganite hands-on thingies



what i've learned about statistics

learning: textbooks/lectures are useful but personally i prefer to just *play* and *do*, if you're similar then rest assured this is still a good way to learn! you are "smart" enough

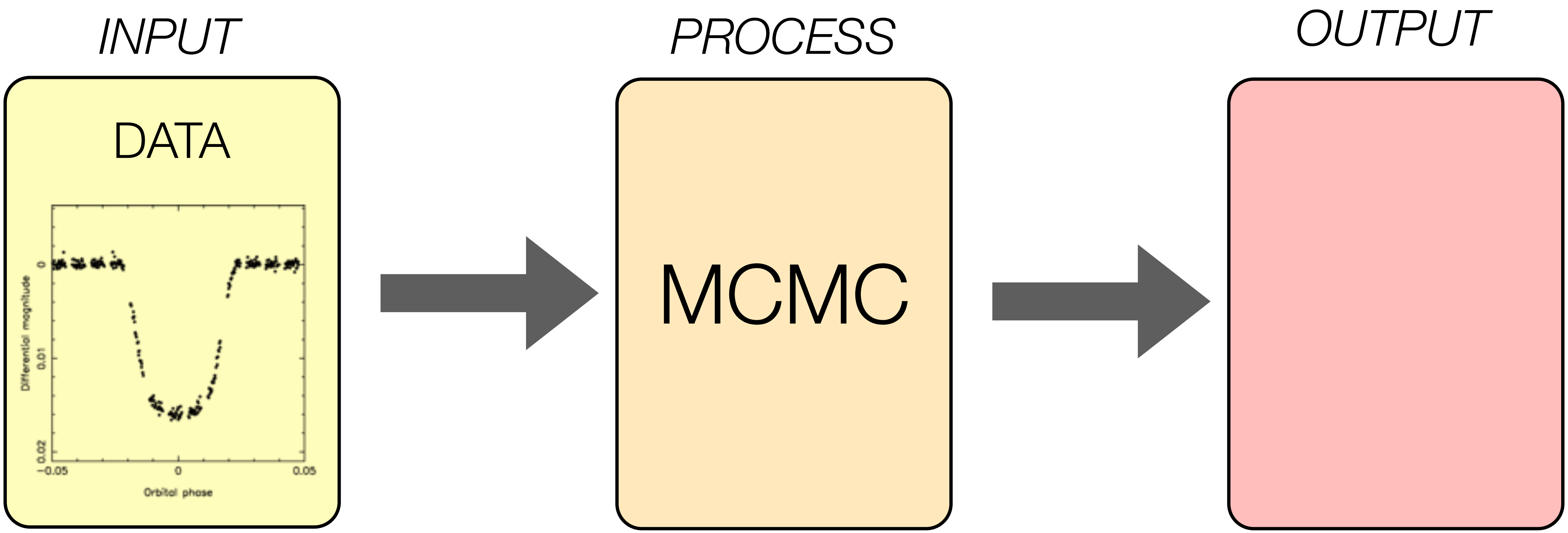
community: astrostatistics is a small but rapidly growing field, many workshops now that I didn't have access to!

credibility: be warned that many respectable astronomers literally say Bayesian statistics is black magic

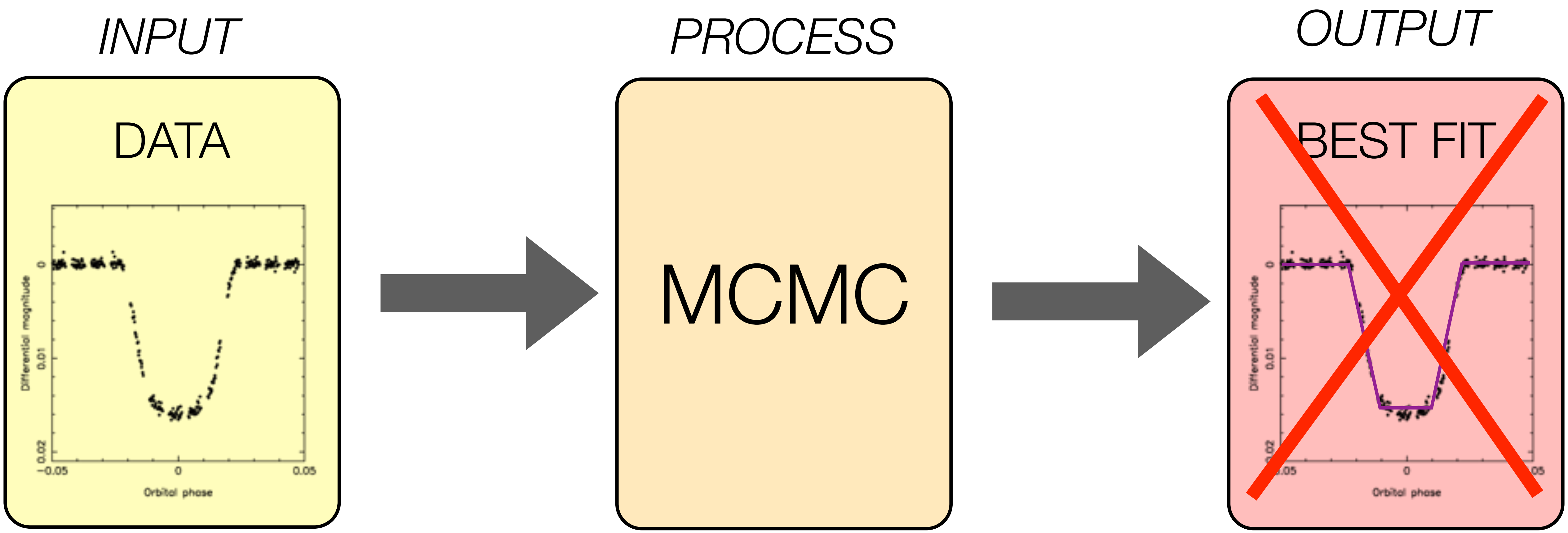
why?: i think of statistics as a means to an end, rather than the end itself, a way to answer astro questions

you: (i bet) you are all more knowledgeable about statistics than I was, i've just learned on the job and learned from many of the amazing lecturers here via papers and talks - so make sure you meet them all!

What is the product of MCMC?



What is the product of MCMC?

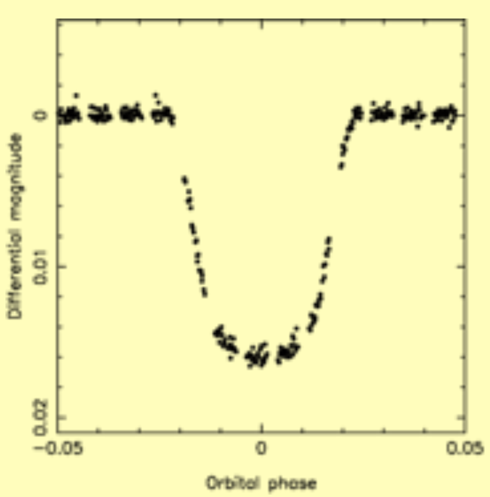


by-product of the MCMC is a reasonable estimate of the best fit, but that's really not it's *raison d'être*

What is the product of MCMC?

INPUT

DATA

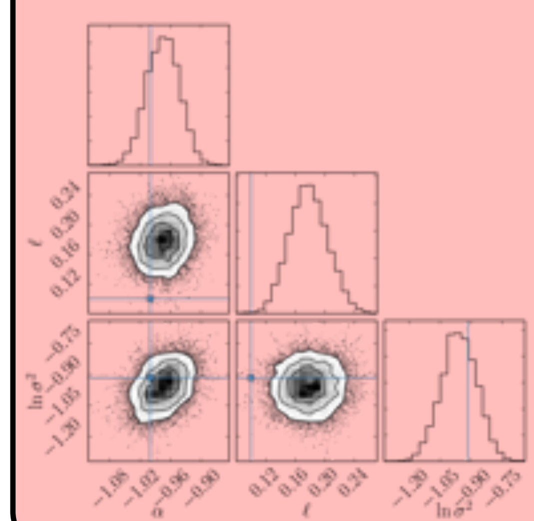


PROCESS

MCMC

OUTPUT

POSTERIOR



and really by this I mean a set of samples from the posterior

what is: ***joint a-posteriori probability distribution*** **("posterior")**

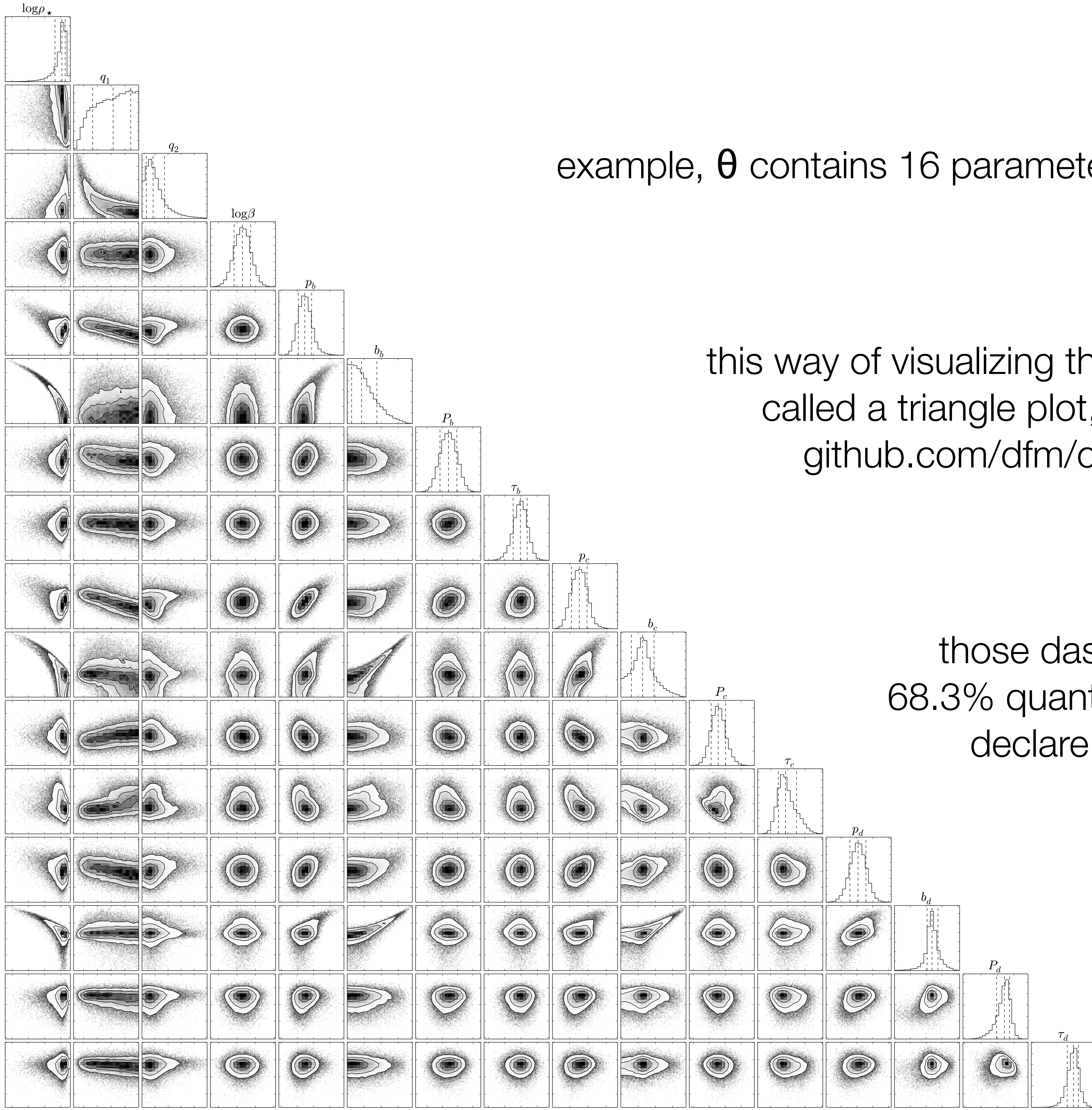
the (joint) probability distribution of some **<u>parameters of interest, θ</u>**, conditioned upon some **<u>data, $\mathscr{D}$</u>** and a **<u>model/hypothesis, $\mathscr{M}$</u>**

basically what's the credible range of your model parameters allowed by your data

(not strictly correct, but OK to think of this way)

you can also think of...
posterior = what you think the parameters are **post** using data
prior = what you think the parameter **prior** to using data

prior belief —*data*→ posterior belief



example, θ contains 16 parameters

this way of visualizing the posteriors is called a triangle plot, check out github.com/dfm/corner.py

those dashed lines are the 68.3% quantiles, which we often declare as θ_1=2.0±0.1

Kipping et al. (2016)

prior belief $\xrightarrow{\text{data}}$ posterior belief

likelihood, $\mathscr{L}$

prior, π

posterior, $\mathscr{P}$

$$P(\boldsymbol{\Theta}|\mathcal{D},\mathscr{M}) = \frac{P(\mathcal{D}|\boldsymbol{\Theta},\mathscr{M})P(\boldsymbol{\Theta}|\mathscr{M})}{P(\mathcal{D}|\mathscr{M})}$$

evidence, ***Z***, (marginal likelihood)

[if you want this for model selection, do nested sampling, not MCMC]

normalization factor, doesn't depend on $\boldsymbol{\Theta}$

$$P(\mathcal{D}|\mathscr{M}) = \int P(\mathcal{D}|\boldsymbol{\Theta},\mathscr{M})P(\boldsymbol{\Theta}|\mathscr{M})\, d^N\boldsymbol{\Theta}$$

prior belief $\xrightarrow{\textit{data}}$ posterior belief

likelihood, $\mathscr{L}$

prior, π

posterior, $\mathscr{P}$

$$\mathrm{P}(\boldsymbol{\Theta}|\mathcal{D},\mathscr{M}) \propto \mathrm{P}(\mathcal{D}|\boldsymbol{\Theta},\mathscr{M})\,\mathrm{P}(\boldsymbol{\Theta}|\mathscr{M})$$

in MCMC, we are just trying to get the posterior, the normalization factor makes no difference to that so ignore it prior belief —*data*→ posterior belief

likelihood,

prior, π

posterior,

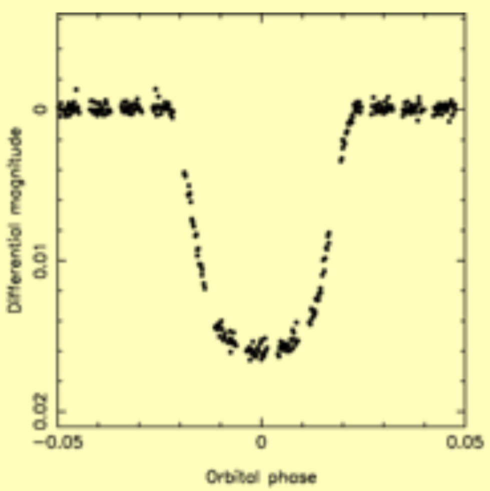
$P(\qquad \qquad P(\mathcal{D}|\mathbf{\Theta},\mathscr{M})P(\mathbf{\Theta}|\mathscr{M})$

in MCMC, we are just trying to get the posterior, the normalization factor makes no difference to that so ignore it

a common sin by MCMC'ers is to pay little attention to the prior... I'll come back to this next lecture *let's expand this out...*

INPUT

DATA

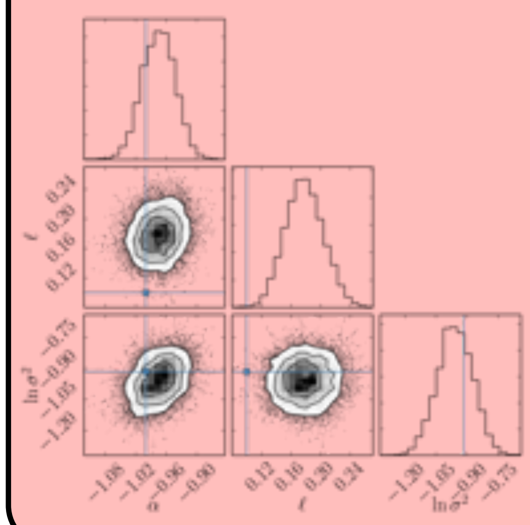


PROCESS

MCMC

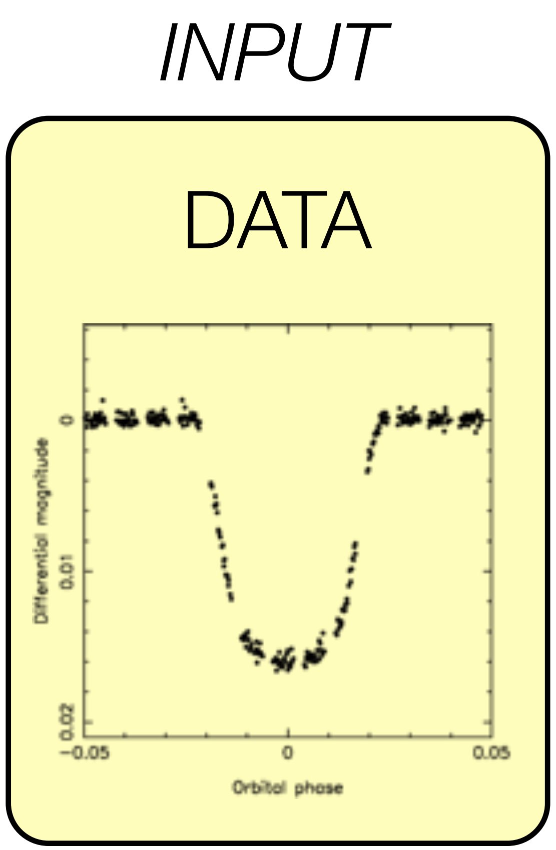
OUTPUT

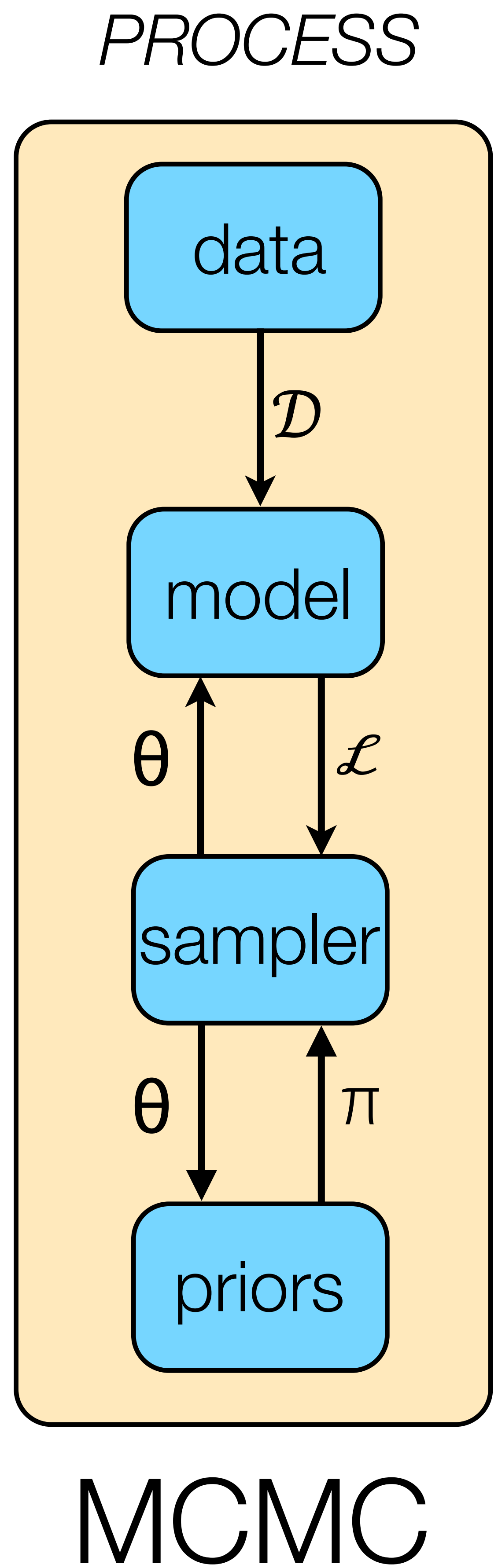
POSTERIOR

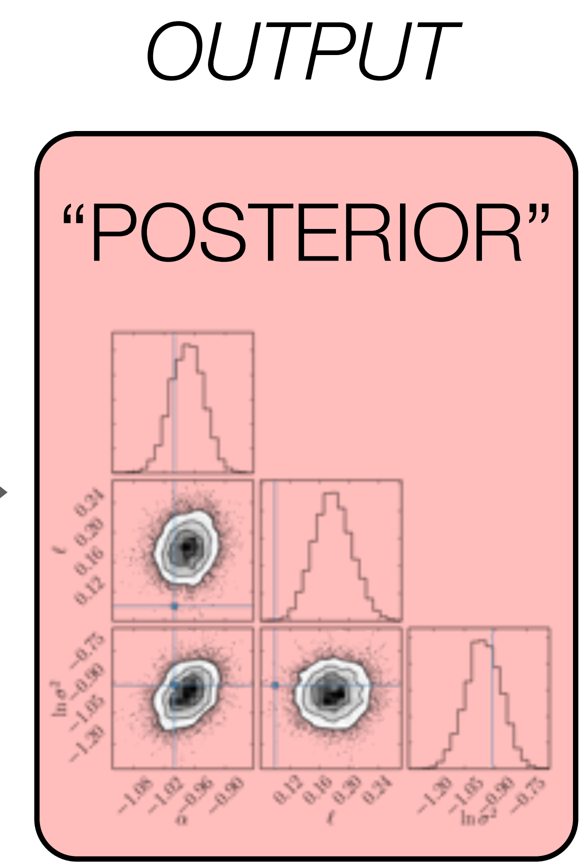


the sampler "guesses" different θ vectors, calculates the posterior probability of that guess, and then makes small jumps

actually the point of the sampler is to make intelligent guesses with high posterior probabilities







so we need...

some data
a model
a sampler
an equation for the likelihood
an equation for the prior

so we need...

some data
a model
a sampler
an equation for the likelihood
an equation for the prior

so we need...

some data

a model

a sampler

an equation for the likelihood

an equation for the prior

likelihood, $\mathscr{L}$

observations are perturbed by stochastic noise

$y_{obs} = y_{true} + \boldsymbol{\varepsilon}$

we never really know the true noise, but often we can make a good approximation, e.g. normally distributed (“white”)

likelihood, $\mathscr{L}$

residuals of data - model

just the pdf of a normal

$$P(\mathcal{D}|\boldsymbol{\Theta},\mathscr{M}) = \prod_{i=1}^{N} \frac{\exp(-\frac{1}{2} r_i^2/\sigma_i^2)}{(2\pi)^{1/2}\sigma_i}$$

measurement uncertainty

it’s often more convenient to calculate $\log\mathscr{L}$

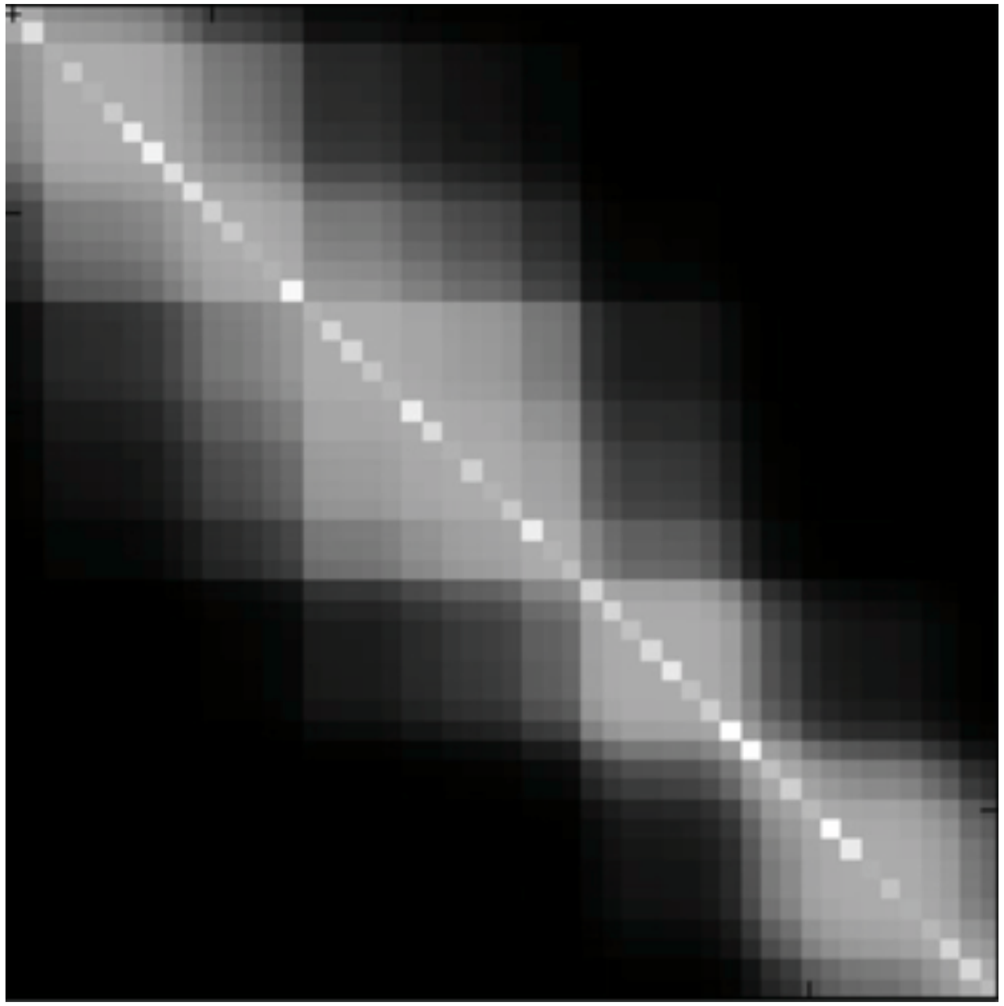
$$\log\mathscr{L} = \tfrac{1}{2} \sum_{i=1}^{N} -\log(2\pi) - \log(\sigma_i^2) - r_i^2/\sigma_i^2 \quad = \chi^2$$

if σ_i=constant

$\log\mathscr{L} = c - \frac{1}{2}\chi^2$

you don’t have assume uncorrelated errors, for example could use a Gaussian process likelihood...

$$\mathrm{P}(\mathcal{D}|\boldsymbol{\Theta},\mathscr{M}) = -\frac{1}{2}\,\mathbf{r}^{\mathrm{T}}\,\mathbf{C}^{-1}\,\mathbf{r} - \frac{1}{2}\log\det\mathbf{C} - \frac{N}{2}\log 2\,\pi$$



check out https://speakerdeck.com/dfm/an-astronomers-introduction-to-gaussian-processes

so we need...

some data
a model
a sampler
an equation for the likelihood
an equation for the prior

so we need...

some data
a model
a sampler
an equation for the likelihood
an equation for the prior *next lecture*

so we need...

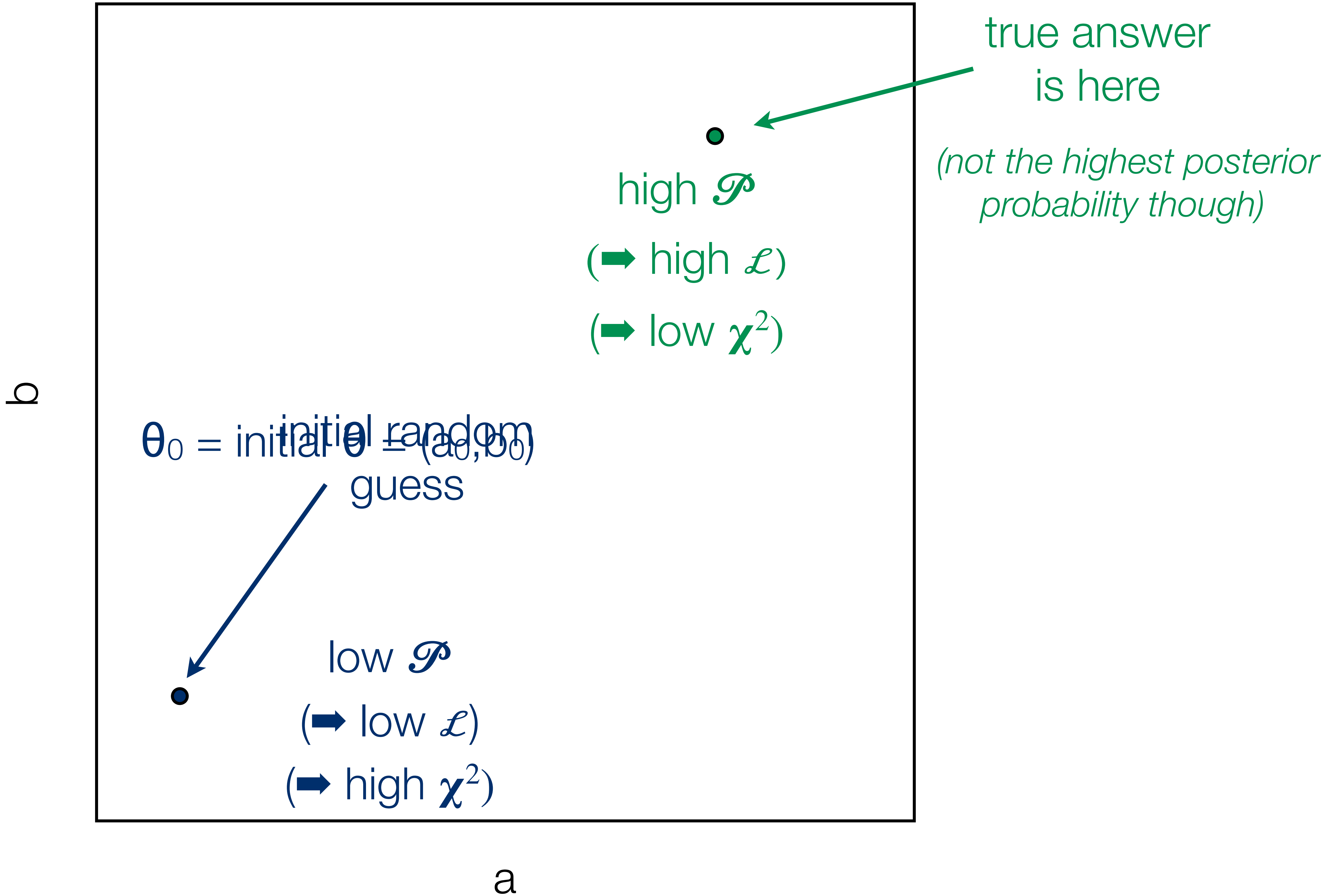
some data
a model
a sampler
an equation for the likelihood
an equation for the prior

simple example:
Metropolis (1953)
algorithm

next lecture

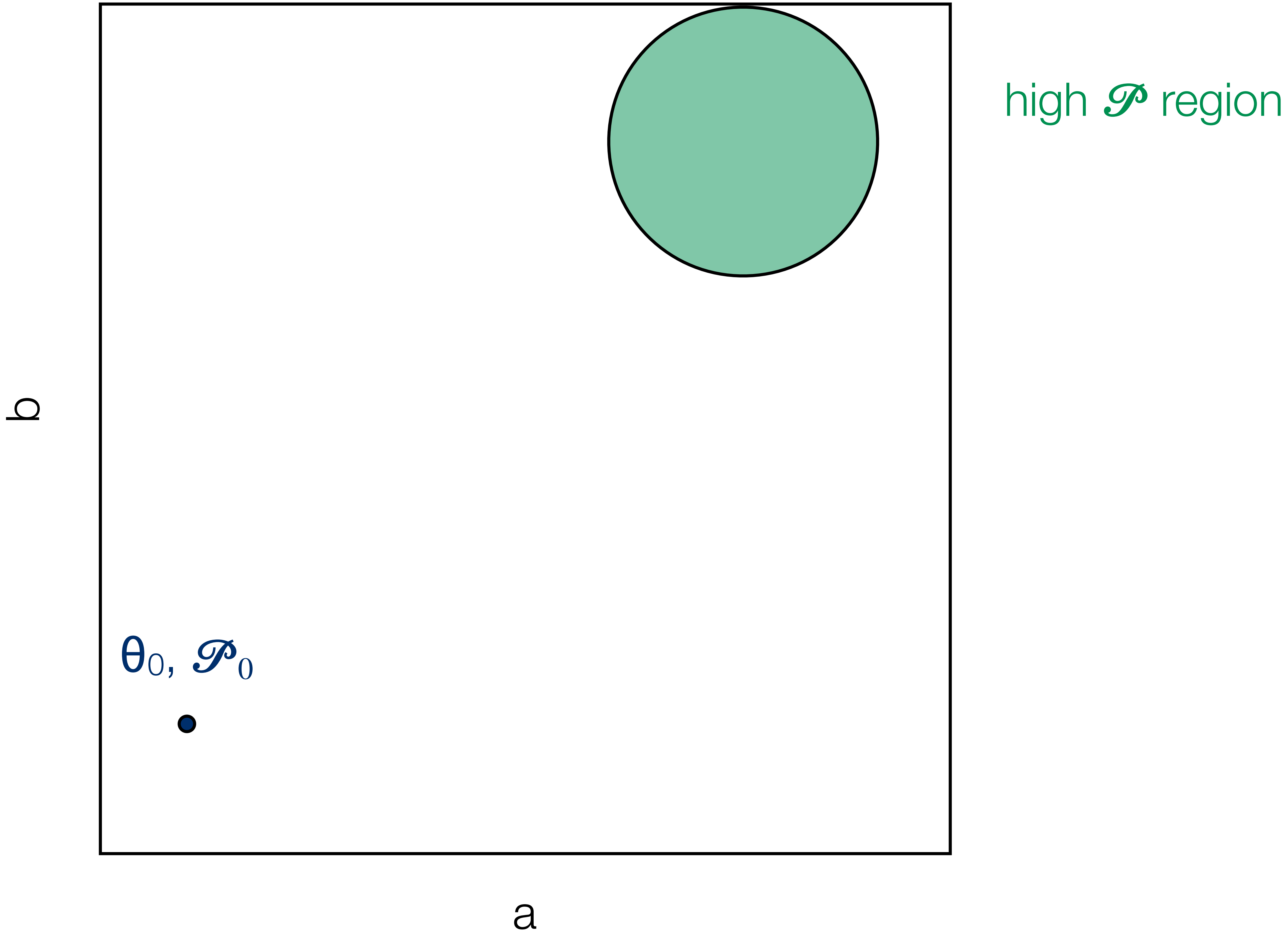
1. define a function for $\mathscr{L}$ & π and thus $\mathscr{P}$

2. define an initial guess for θ from π

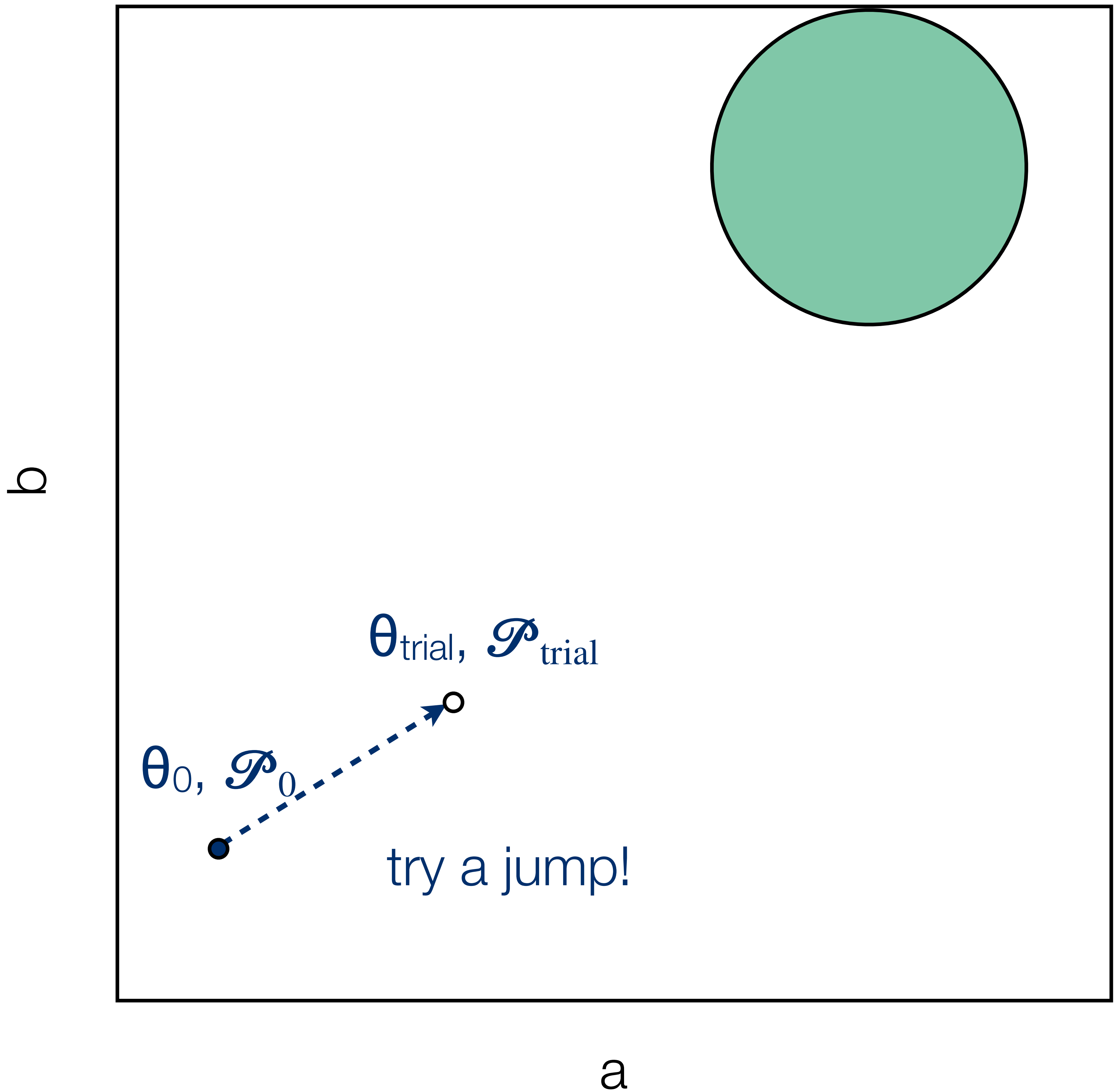


1. define a function for $\mathscr{L}$ & π and thus $\mathscr{P}$

2. define an initial guess for θ from π



3. try a jump in θ



high $\mathscr{P}$ region

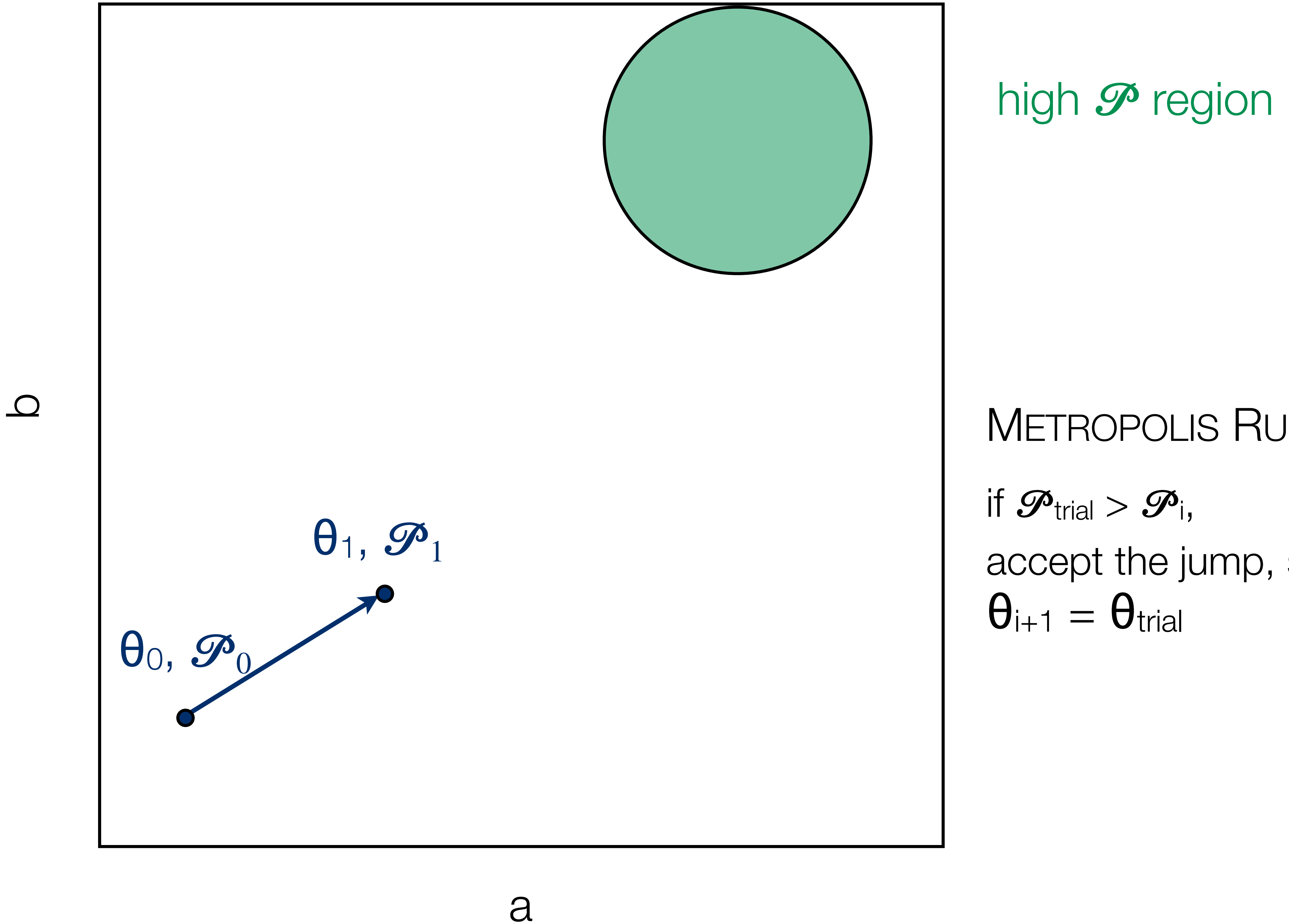
Metropolis Rule

if $\mathscr{P}_{trial} > \mathscr{P}_i$,

accept the jump, so

$\theta_{i+1} = \theta_{trial}$

3. try a jump in θ



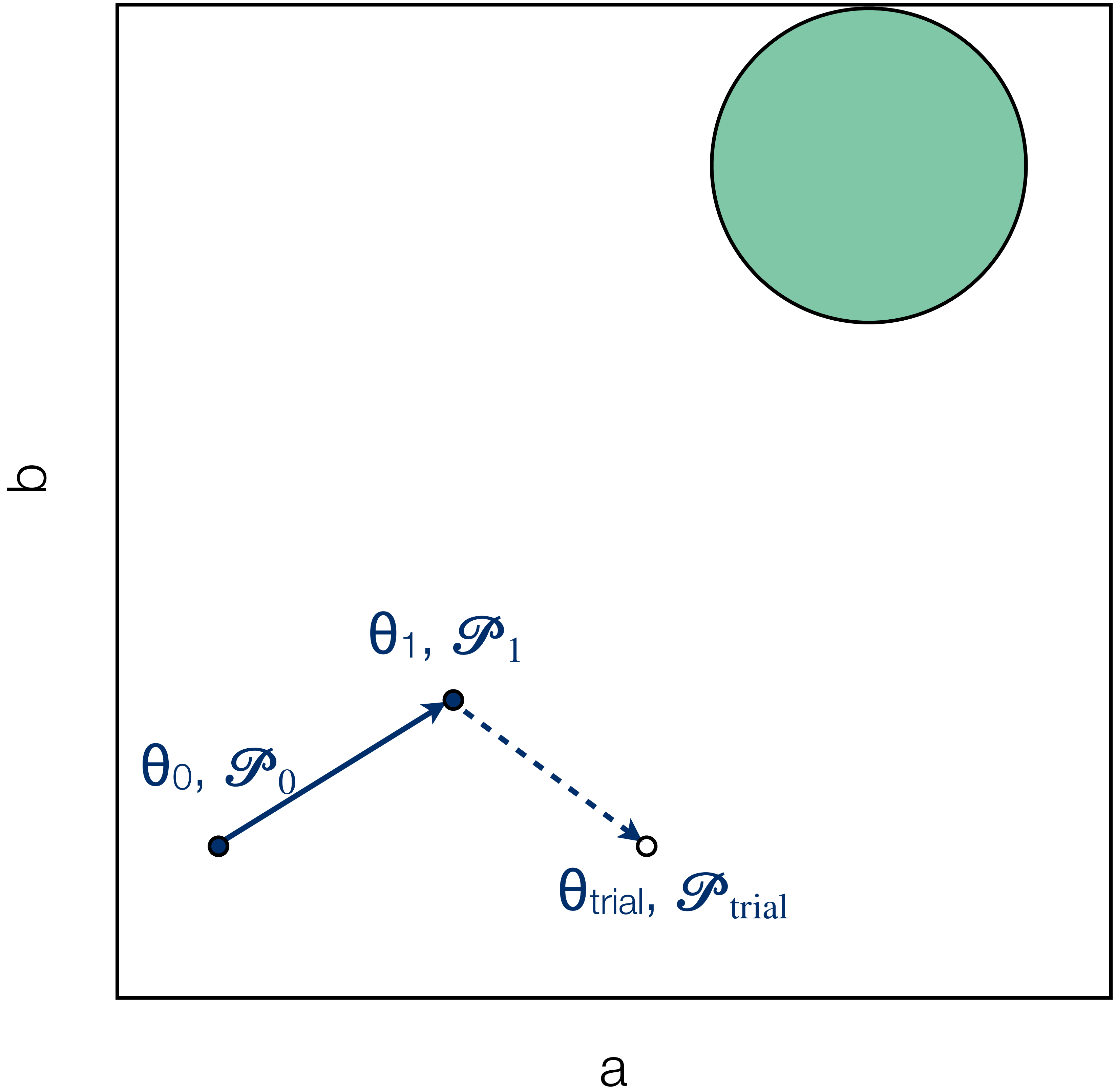
Metropolis Rule

if $\mathscr{P}_{trial} > \mathscr{P}_i$,

accept the jump, so

$\theta_{i+1} = \theta_{trial}$

3. try a jump in θ



high $\mathscr{P}$ region

Metropolis Rule

if $\mathscr{P}_{trial} > \mathscr{P}_i$,

accept the jump, so $\theta_{i+1} = \theta_{trial}$

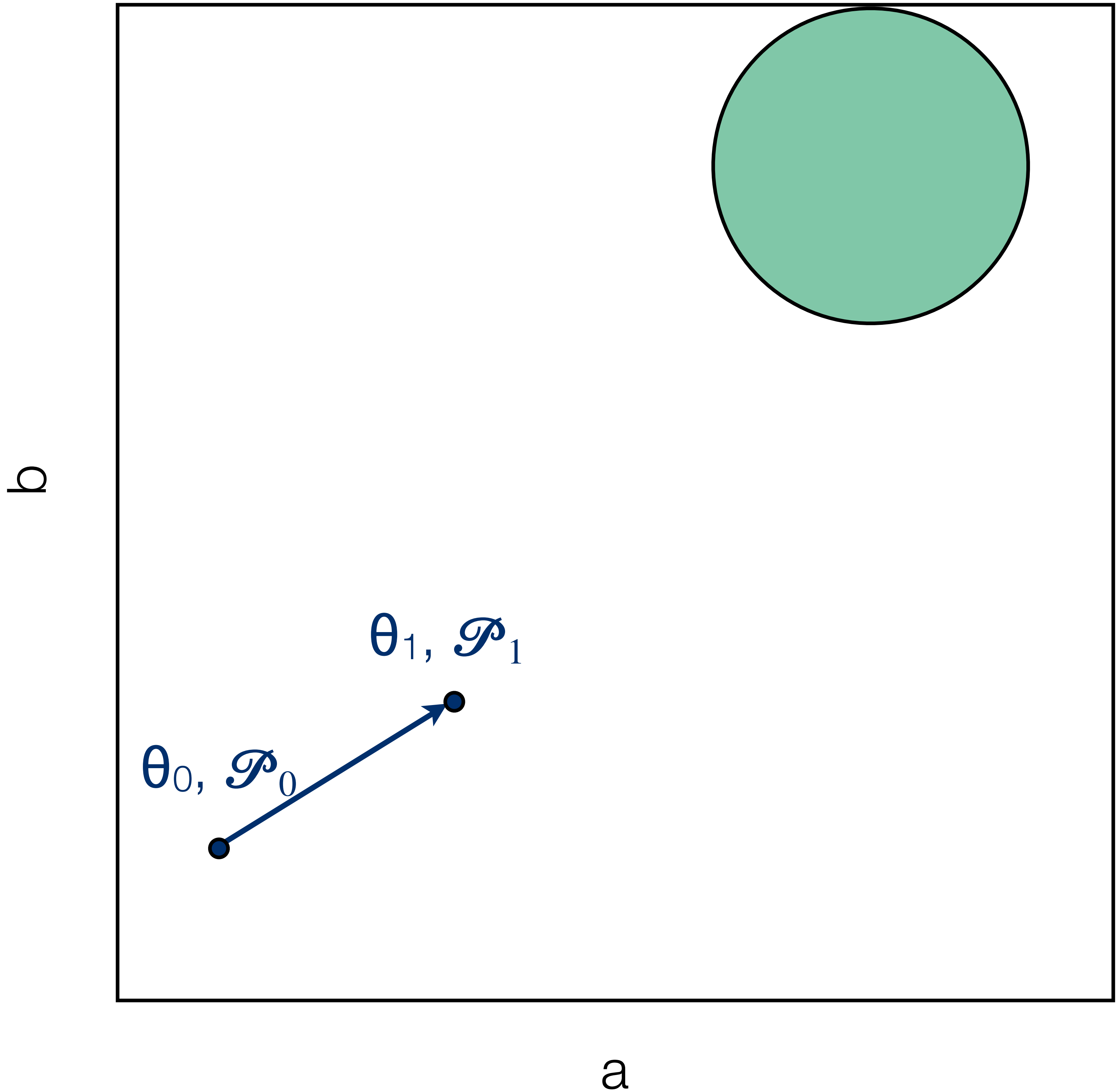
if $\mathscr{P}_{trial} < \mathscr{P}_i$,

accept the jump with probability $\mathscr{P}_{trial}/\mathscr{P}_i$

this is why evidence doesn't matter in MCMC!

3. try a jump in θ

4. accept/reject based on Metropolis Rule



high $\mathscr{P}$ region

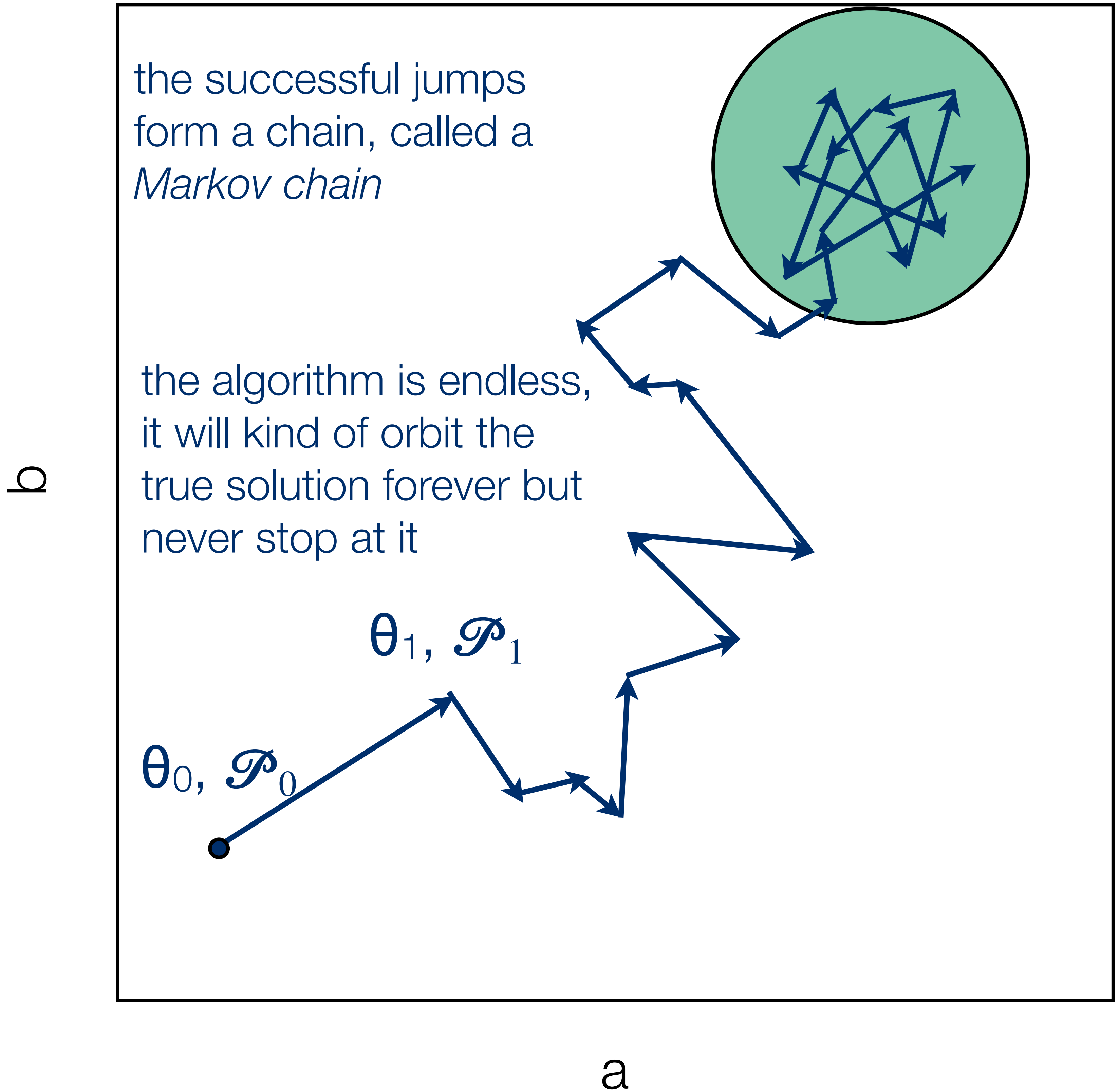
that's it!

METROPOLIS RULE

if $\mathscr{P}_{trial} > \mathscr{P}_i$,
accept the jump, so
$\theta_{i+1} = \theta_{trial}$

if $\mathscr{P}_{trial} < \mathscr{P}_i$,
accept the jump with probability $\mathscr{P}_{trial}/\mathscr{P}_i$

5. keep jumping!



Metropolis Rule

if $\mathscr{P}_{trial} > \mathscr{P}_i$,

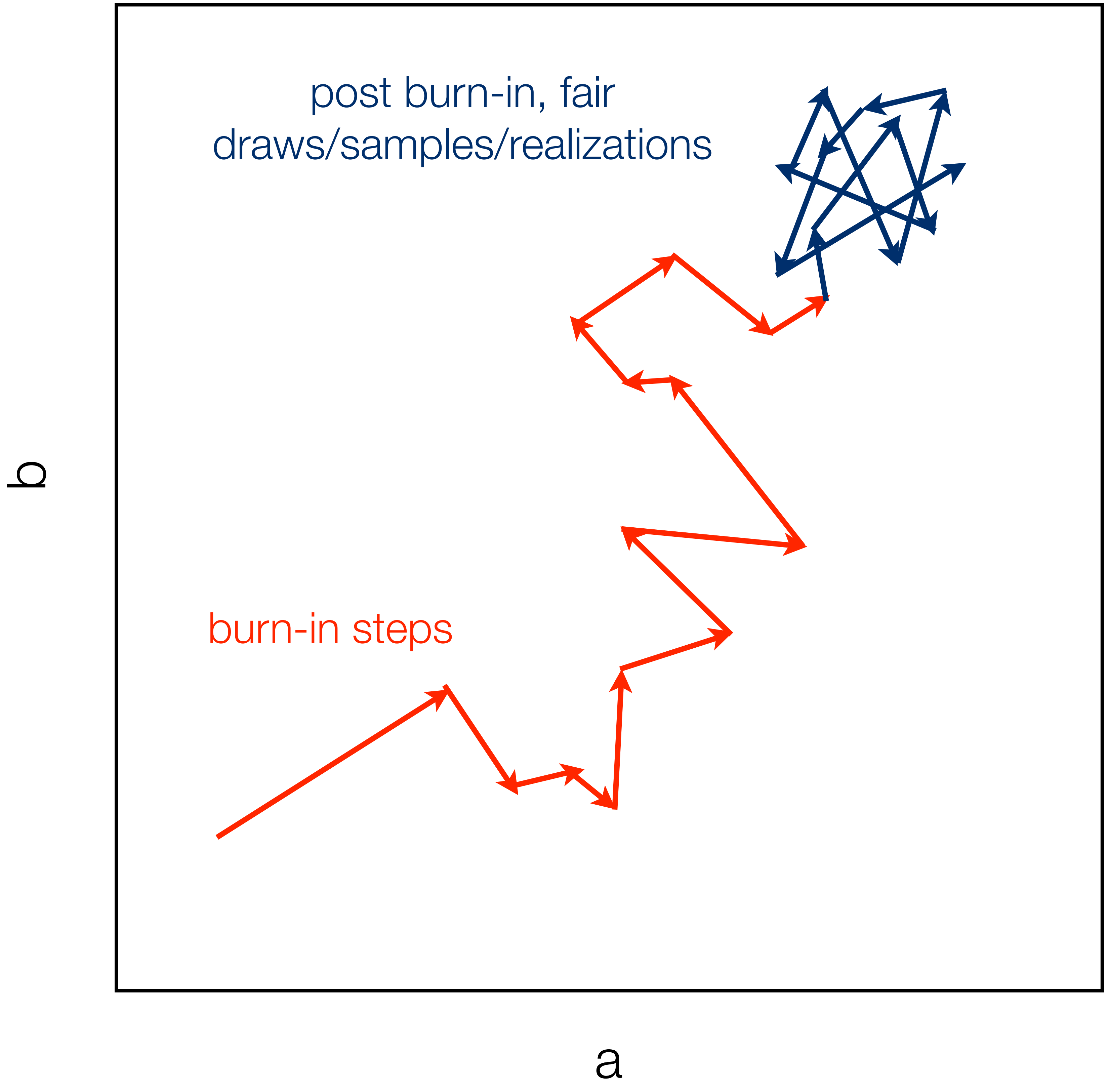
accept the jump, so $\theta_{i+1} = \theta_{trial}$

if $\mathscr{P}_{trial} < \mathscr{P}_i$,

accept the jump with probability $\mathscr{P}_{trial}/\mathscr{P}_i$

5. keep jumping!

6. after you've done many steps, remove burn-in steps



Metropolis Rule

if $\mathscr{P}_{\mathrm{trial}} > \mathscr{P}_{\mathrm{i}}$,

accept the jump, so $\theta_{i+1} = \theta_{\mathrm{trial}}$

if $\mathscr{P}_{\mathrm{trial}} < \mathscr{P}_{\mathrm{i}}$,

accept the jump with probability $\mathscr{P}_{\mathrm{trial}}/\mathscr{P}_{\mathrm{i}}$

general case

METROPOLIS RULE

if $\mathscr{P}_{trial} > \mathscr{P}_i$,

accept the jump, so $\theta_{i+1} = \theta_{trial}$

if $\mathscr{P}_{trial} < \mathscr{P}_i$,

accept the jump with probability $\mathscr{P}_{trial}/\mathscr{P}_i$

someone ignoring priors

METROPOLIS RULE

if $\mathscr{L}_{trial} > \mathscr{L}_i$,

accept the jump, so $\theta_{i+1} = \theta_{trial}$

if $\mathscr{L}_{trial} < \mathscr{L}_i$,

accept the jump with probability $\mathscr{L}_{trial}/\mathscr{L}_i$

someone ignoring priors and assuming normal errors

METROPOLIS RULE

if $\chi^2_{trial} < \chi^2_i$,

accept the jump, so $\theta_{i+1} = \theta_{trial}$

if $\chi^2_{trial} > \chi^2_i$,

accept the jump with probability $\exp(-\Delta\chi^2/2)$

burn-in point can be spotted by eye...

MCMC algorithm

1. define a function for $\mathscr{L}$ & π and thus $\mathscr{P}$
2. define an initial guess for θ from π
3. try a jump in θ
4. accept/reject based on Metropolis Rule
5. keep jumping!
6. after you've done many steps, remove burn-in steps

How to make jumps (proposals)?

simplest thing is to use a normal distribution

let $\theta_{trial} = \theta_i + \mathcal{N}(0, \Delta\theta)$

so draw a random number from a normal distribution with stdev = “jump scale”

ok...so how do I choose jump scale, $\Delta\theta$?!

That’s tricky, too small and it will take forever, too big and you will overshoot. Experiment, and ideally tune to a number which leads to a 10-70% acceptance rate

(you have to do this for each dimension!)

some checks to do...

caveat: for each of these, there are no single right answers that always work, always inspect your chains, but here are some useful tips...

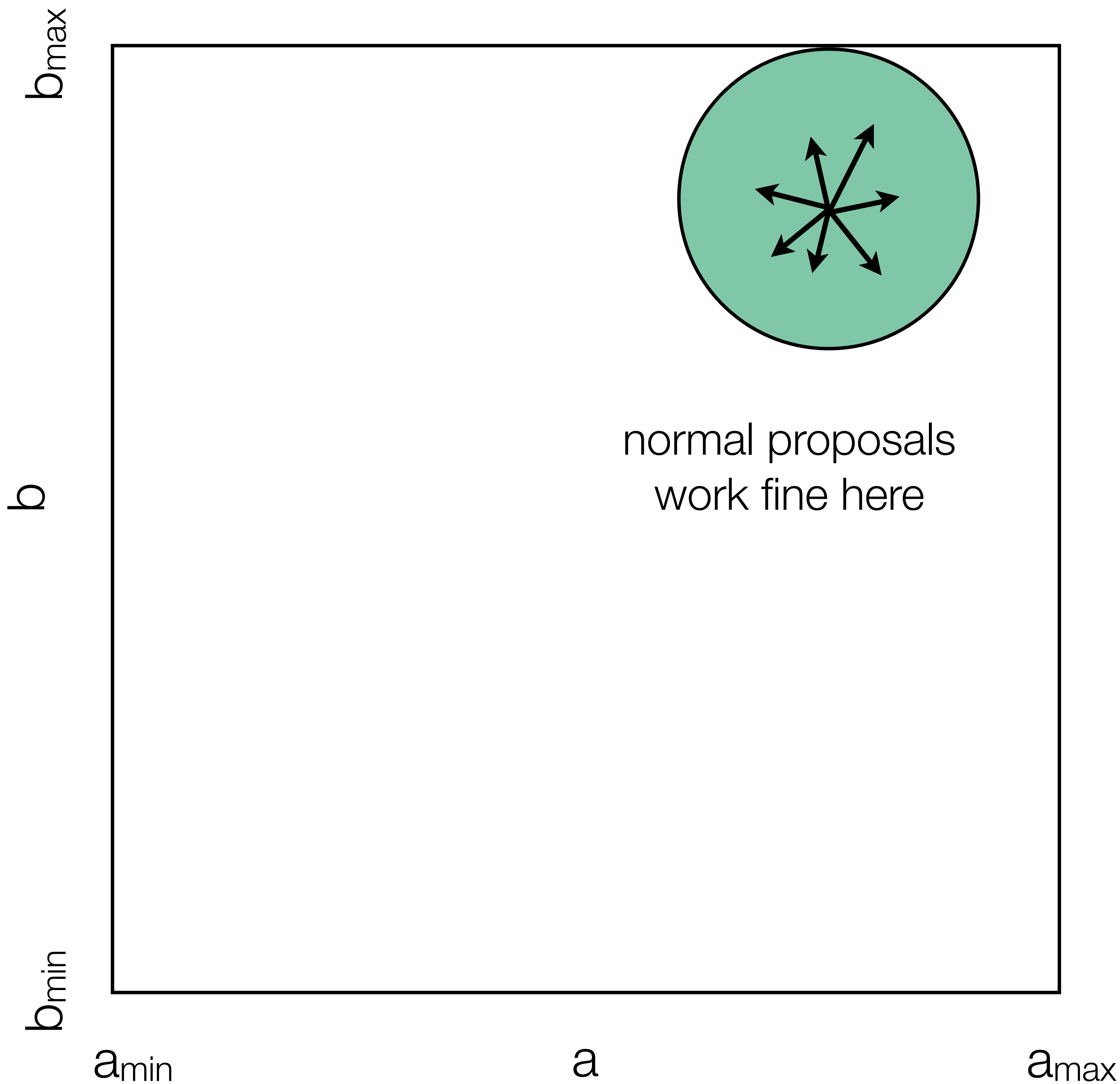
Burn-in: when the chain's likelihood exceeds the median likelihood of the entire chain, demarks burn-in point

Mixing: effective length of the chain should be at least a few hundred, ideally thousands (each eff length defines a part of the chain which is highly auto-correlated, common cutoff is 0.5)

Convergence: Run multiple chains independently and make sure they arrive at the same end point, Gelman-Rubins statistic is a useful check

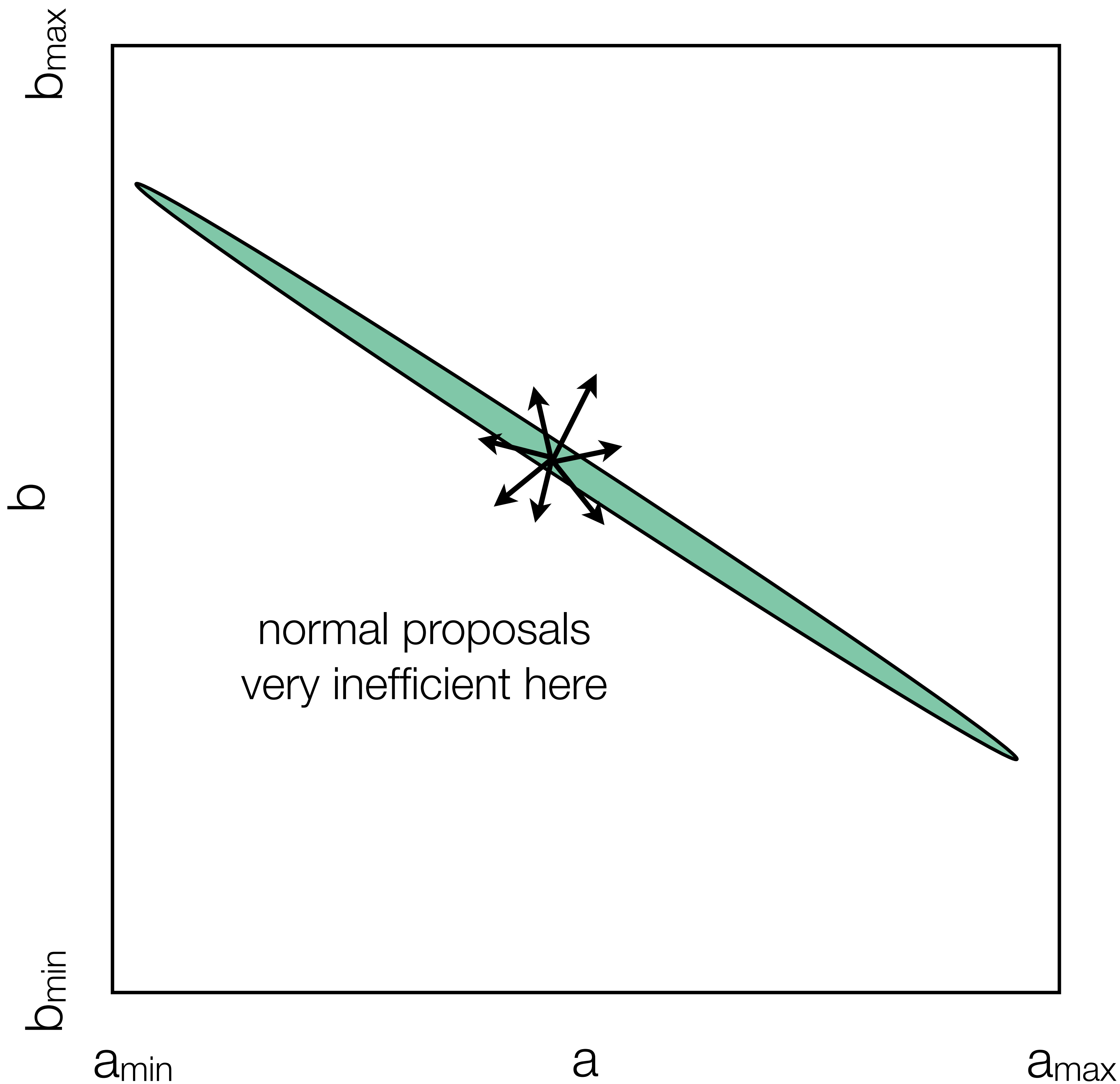
even if you do all that...

...Metropolis can still be a real pain for certain problems



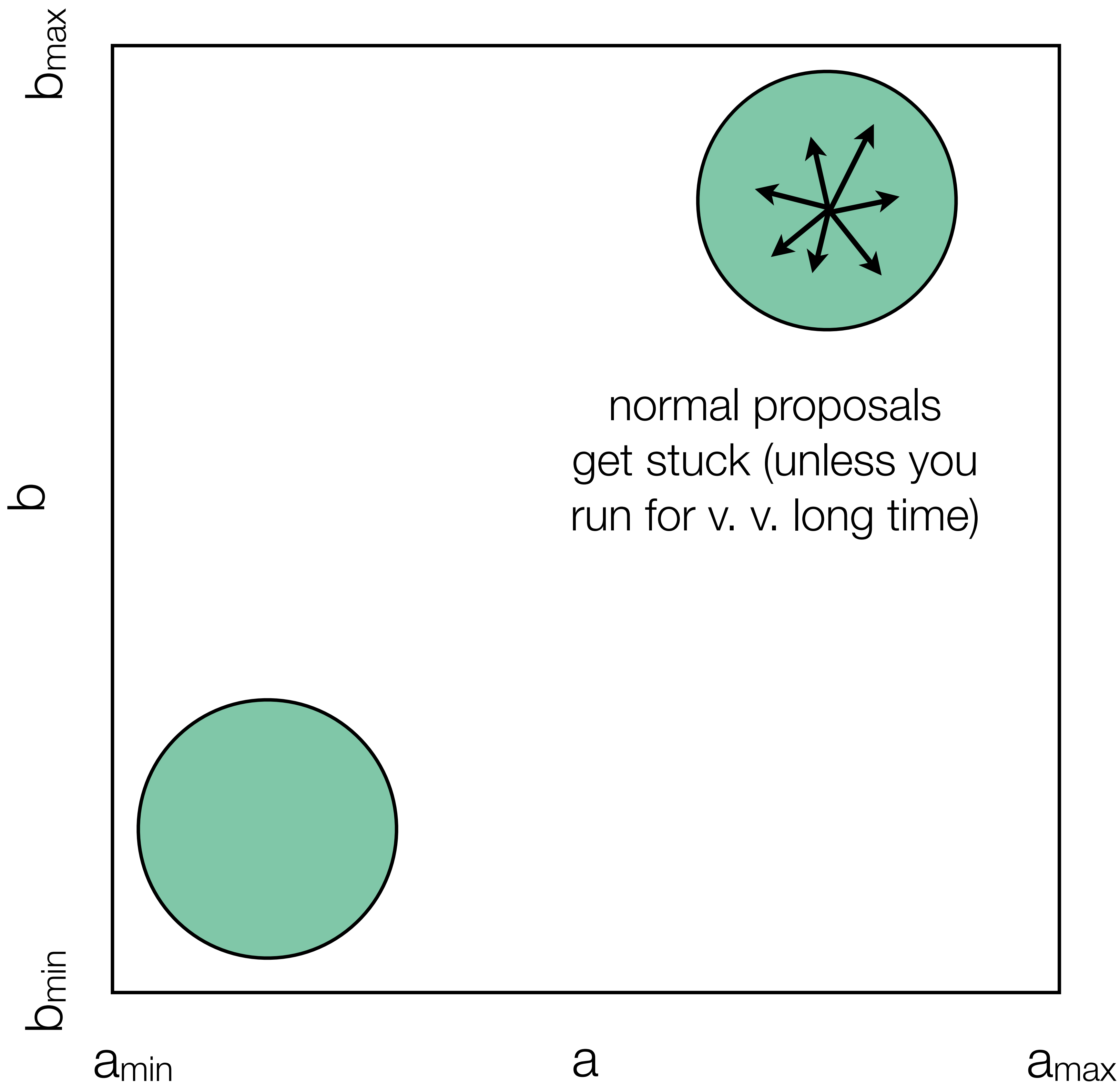
even if you do all that...

...Metropolis can still be a real pain for certain problems



even if you do all that...

...Metropolis can still be a real pain for certain problems



my advise...

write your own Metropolis MCMC, it's a great way to learn

but except for simple problems, it's difficult to know what a good proposal function is, so you will probably want to use a smarter sampler than Metropolis

fortunately there are many more sophisticated techniques available to you...

some examples...

(non-exhaustive! there are hundreds of methods!)

metropolis-hastings

generalization of metropolis to asymmetric proposals

METROPOLIS RULE

accept the jump with probability min(a,1):

$$a = \frac{\mathscr{P}(\theta_{trial})}{\mathscr{P}(\theta_i)}$$

METROPOLIS HASTINGS RULE

accept the jump with probability min(a,1):

$$a = \frac{\mathscr{P}(\theta_{trial})/J(\theta_{trial}|\theta_i)}{\mathscr{P}(\theta_i)/J(\theta_i|\theta_{trial})}$$

Hastings (1970)

simulated annealing

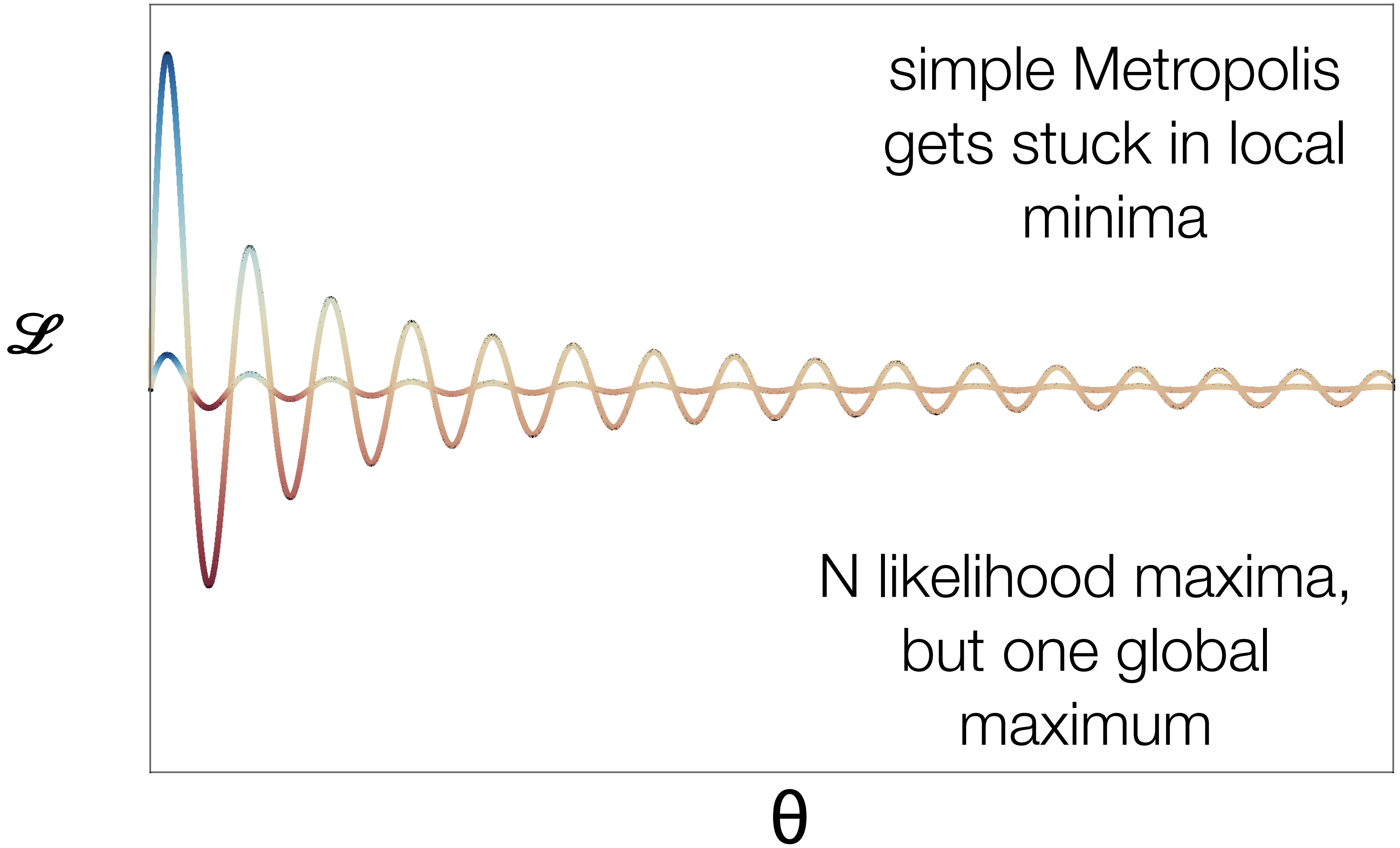
good for multi-modal problems

if $\mathscr{P}_{trial} < \mathscr{P}_i$,
accept the jump with
probability $(\mathscr{P}_{trial}/\mathscr{P}_i)$

→

if $\mathscr{P}_{trial} < \mathscr{P}_i$,
accept the jump with
probability $(\mathscr{P}_{trial}/\mathscr{P}_i)^{1/T}$

usually the jump sizes are increased similarly

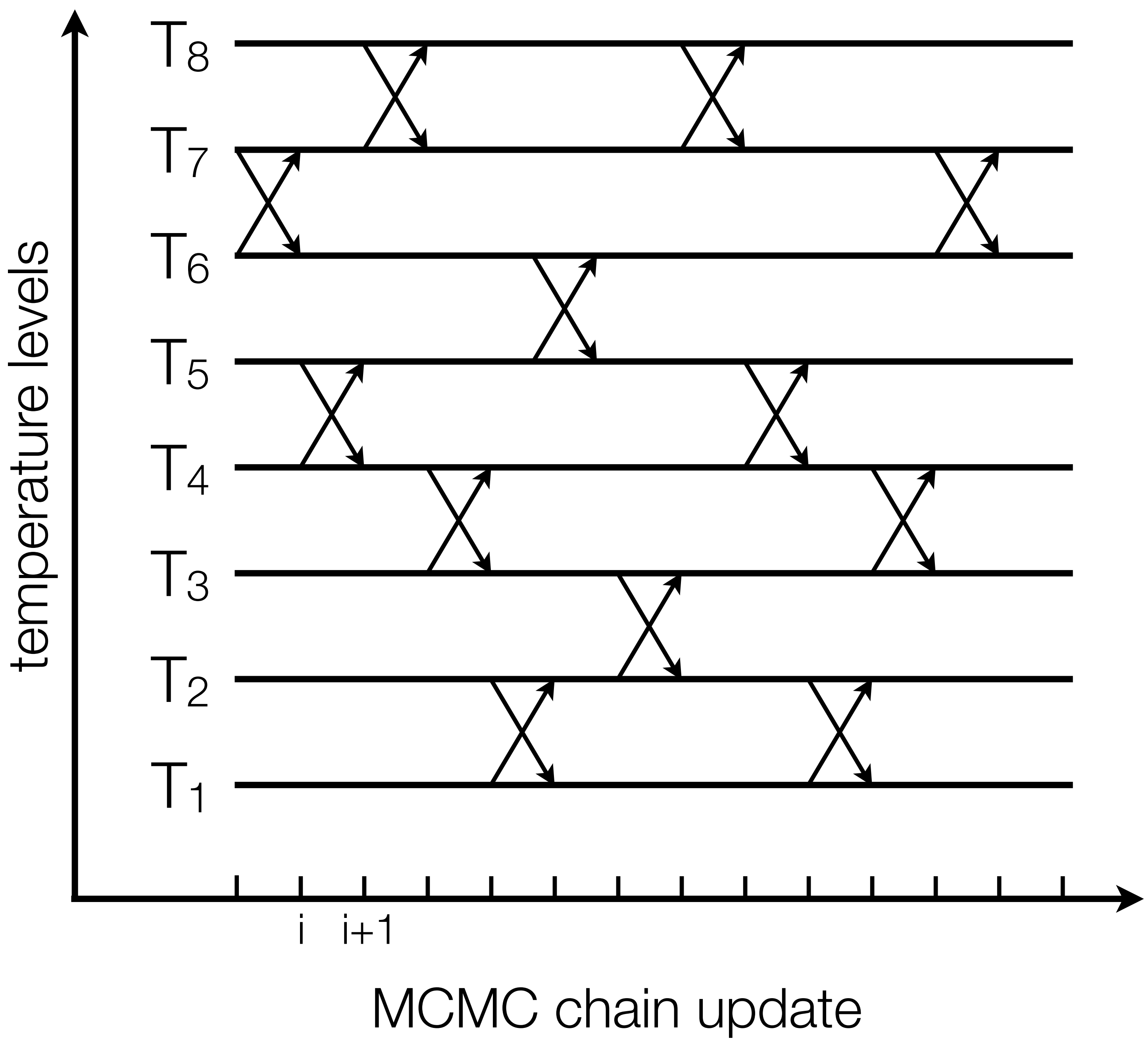


think of it as smoothing out the likelihood space at high temperatures

gradually turn the temperature down until you hit T=1, you can only use samples from that level ("cooling schedule")

parallel tempering

good for multi-modal problems with parallel computing



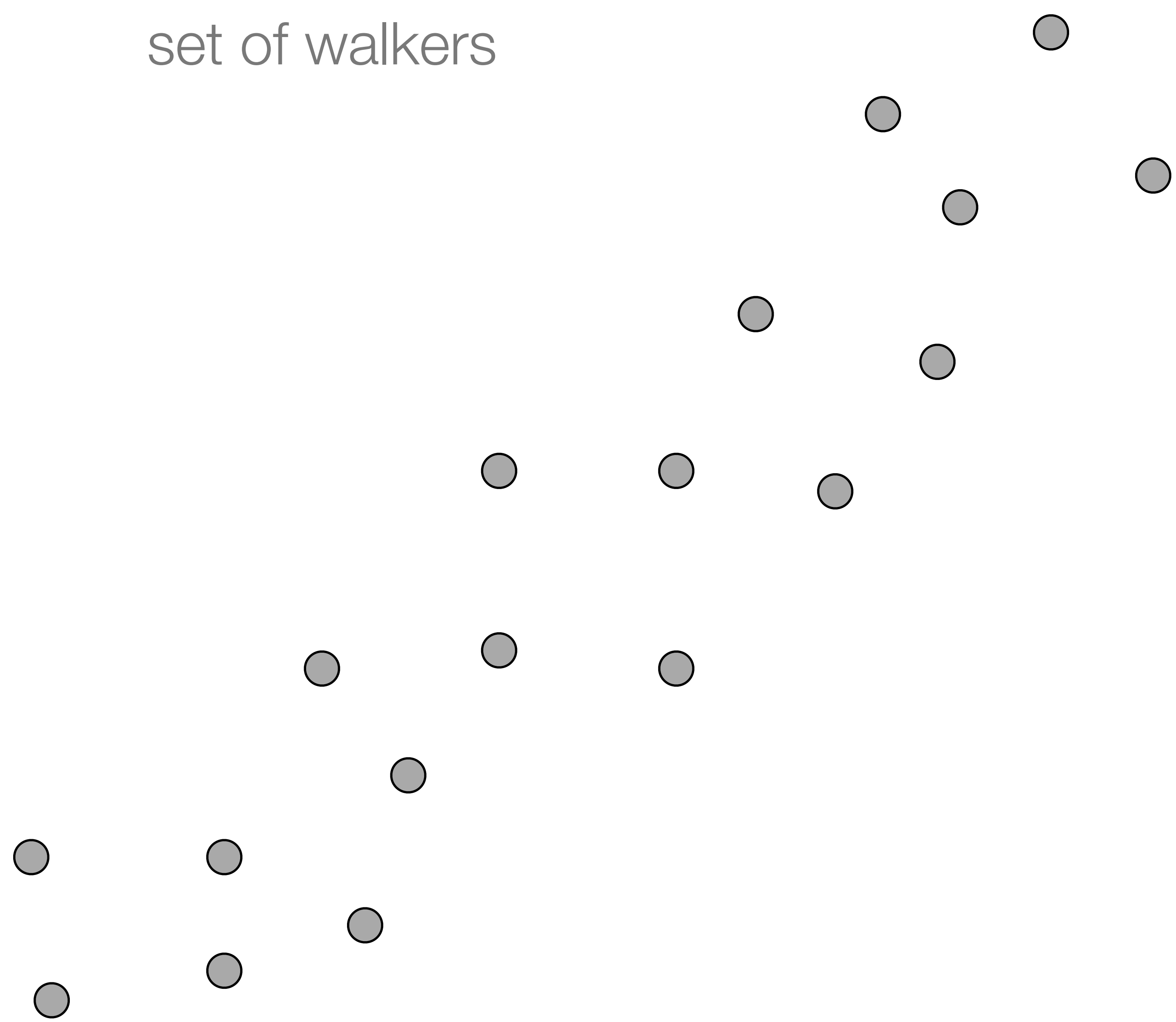
similar to simulated annealing, except temperatures are not run in series but in parallel

at a pre-set step frequency, allow chains to swap

only the lowest chain is used for posterior samples

affine-invariant sampling

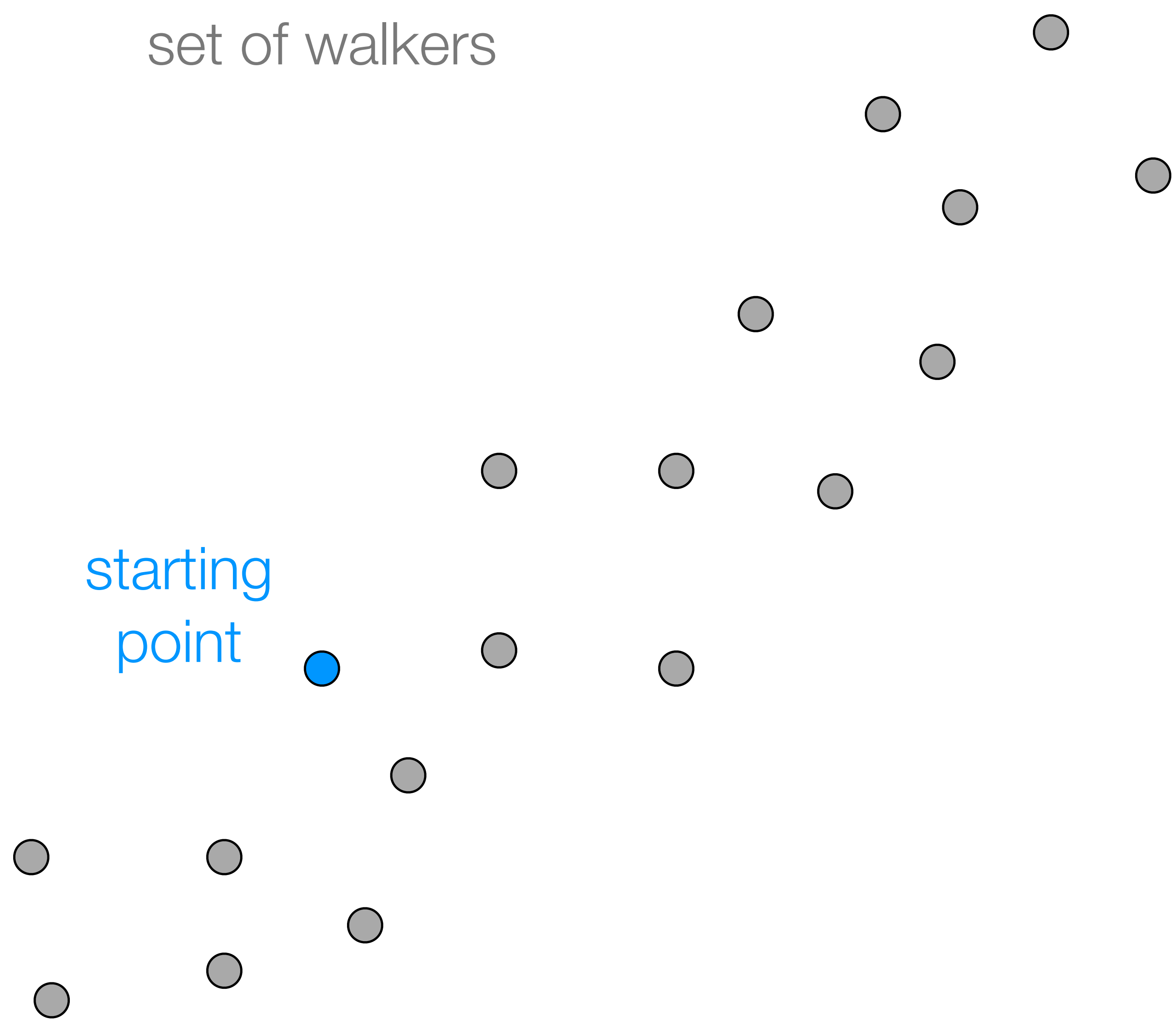
good for multi-modal & correlated problems with parallel computing



Goodman & Weare (2010)

affine-invariant sampling

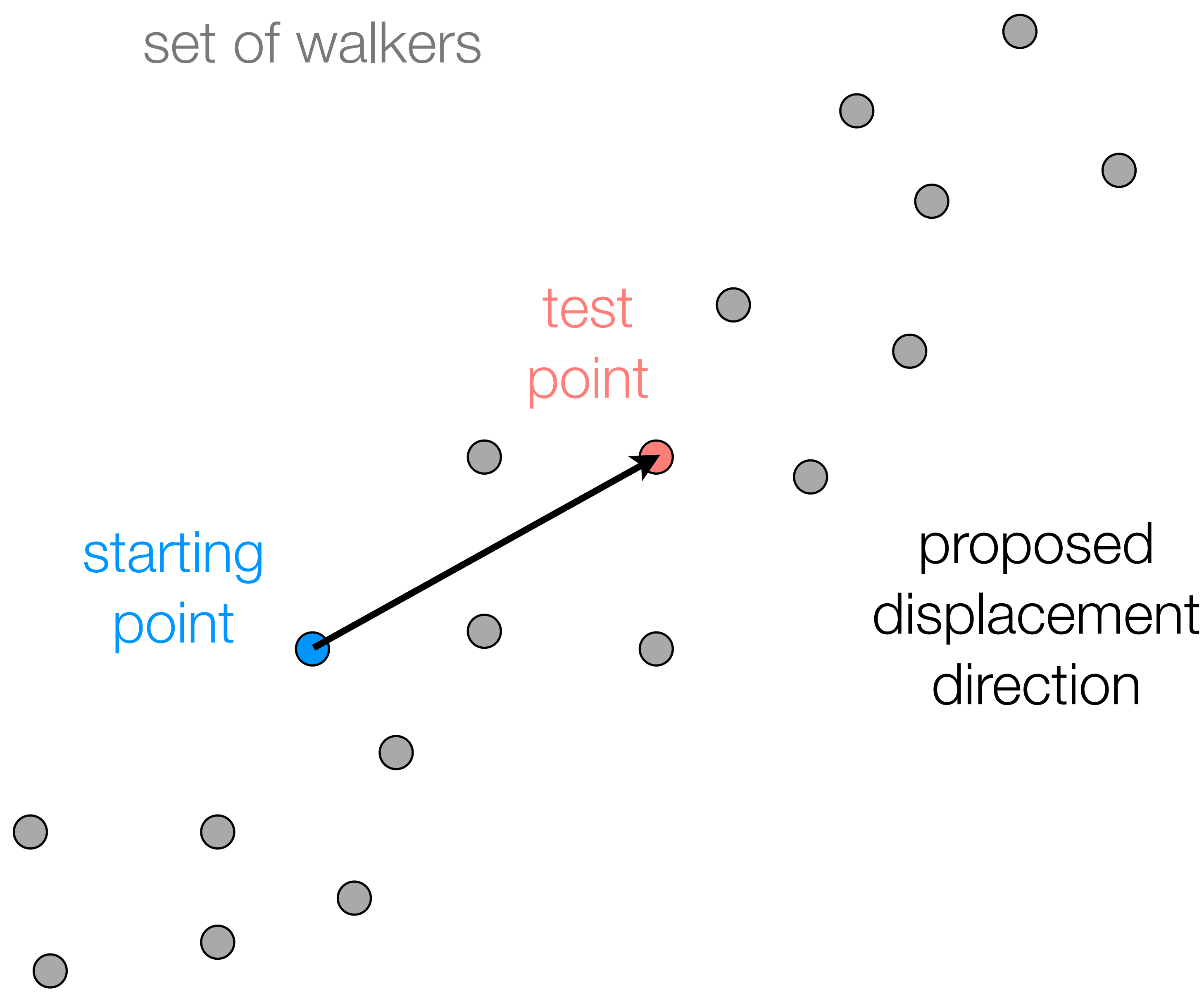
good for multi-modal & correlated problems with parallel computing



Goodman & Weare (2010)

affine-invariant sampling

good for multi-modal & correlated problems with parallel computing

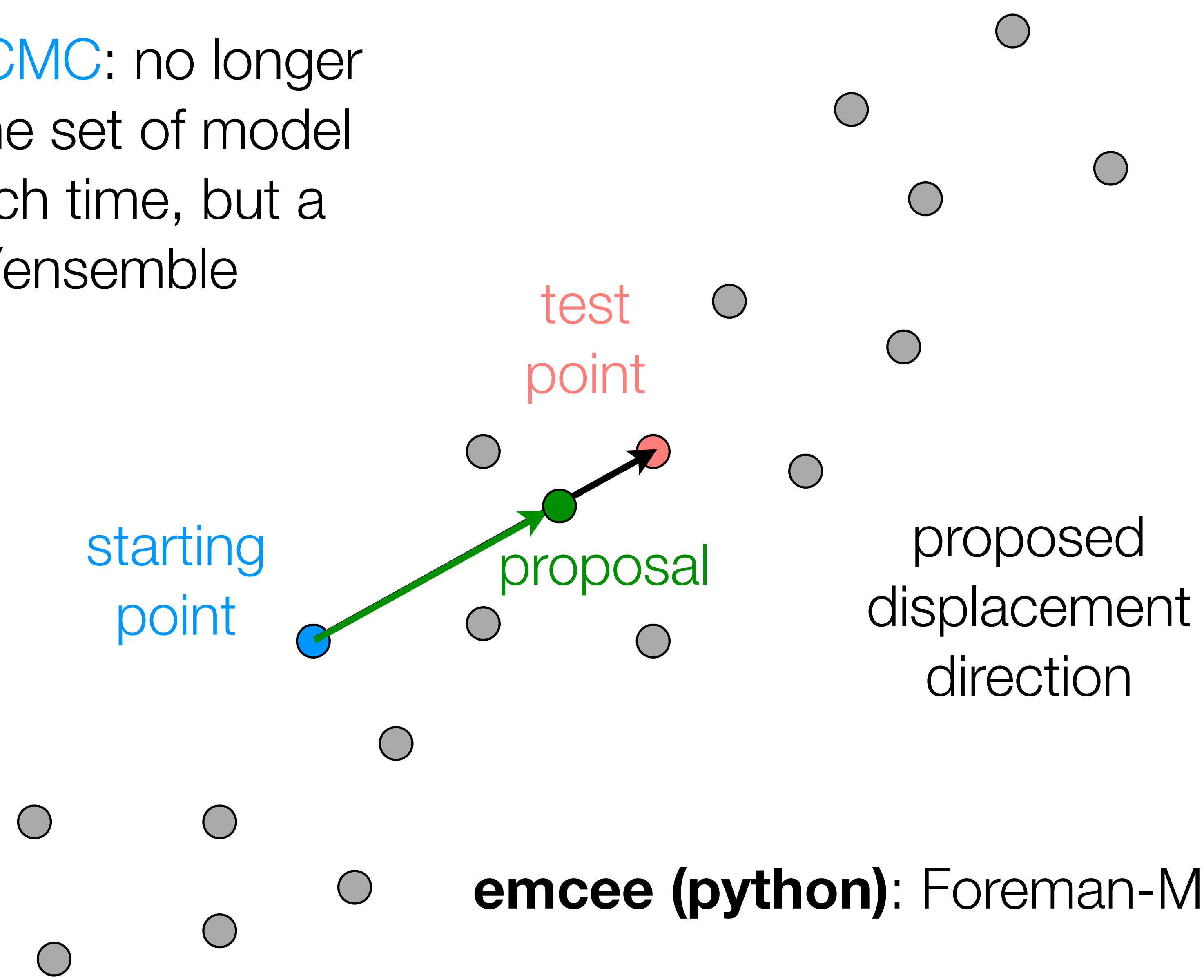


Goodman & Weare (2010)

affine-invariant sampling

good for multi-modal & correlated problems with parallel computing

an ensemble MCMC: no longer just updating one set of model parameters each time, but a generation/ensemble

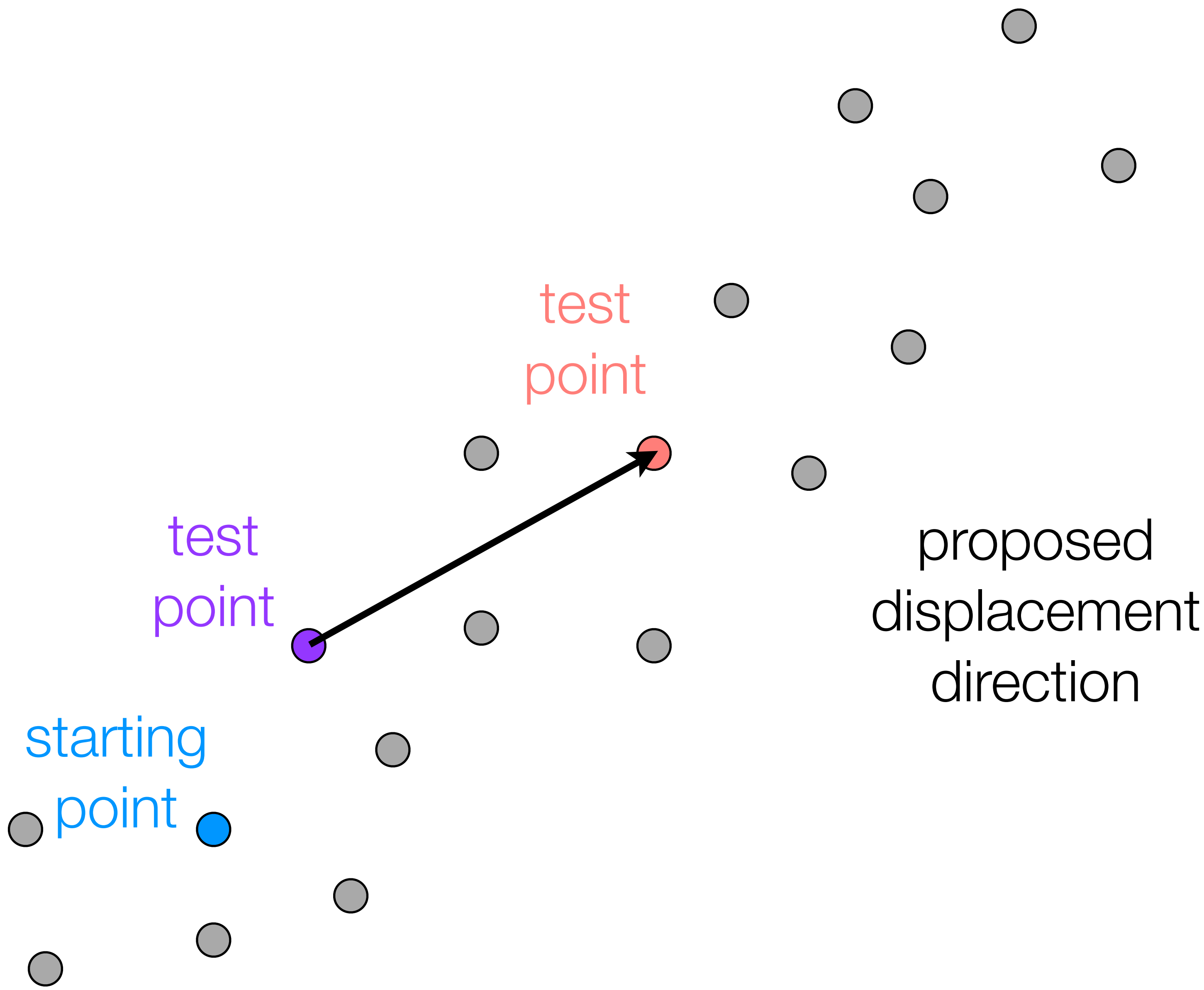


emcee (python): Foreman-Mackey et al. (2013)

Goodman & Weare (2010)

differential evolution

good for multi-modal & correlated problems with parallel computing

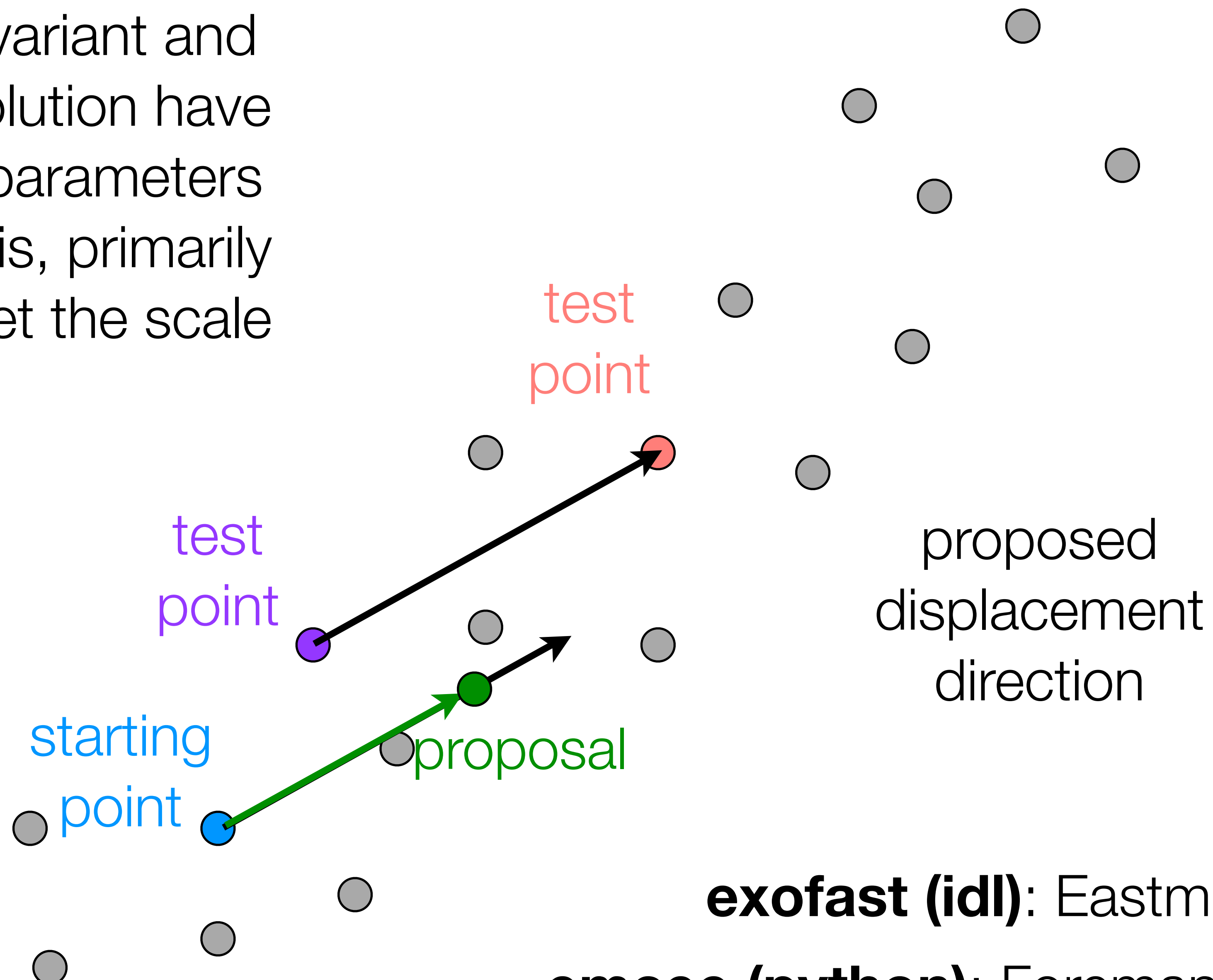


ter Braak (2006)

differential evolution

good for multi-modal & correlated problems with parallel computing

both affine invariant and differential evolution have fewer tuning parameters than Metropolis, primarily need to just set the scale



exofast (idl): Eastman et al. (2012)

emcee (python): Foreman-Mackey et al. (2013)

ter Braak (2006)

getting started

- first make sure you are comfortable with the concepts of priors, likelihood and posteriors
- then try coding up your own MCMC with Metropolis sampling in your favourite language, run on some toy problems
- before choosing a prepackaged MCMC, think about your problem e.g. dimensionality, correlations, likelihood cost, multimodality
- then do some research about "good" algorithms for your problem: literature search, google, Astrostatistics FB group, ask colleagues!
- (if a few options, choose the one you feel like you best understand!)