Lecture 13: Introduction to parameter estimation

Complex Systems 530

Outline

- Today
  - Intro to parameter estimation (for models in general, not just ABMs)
  - Some of the challenges involved in using these tools for ABMs
- Next time: Bayesian & sampling based approaches, intro to MCMC

Connecting Models with Data

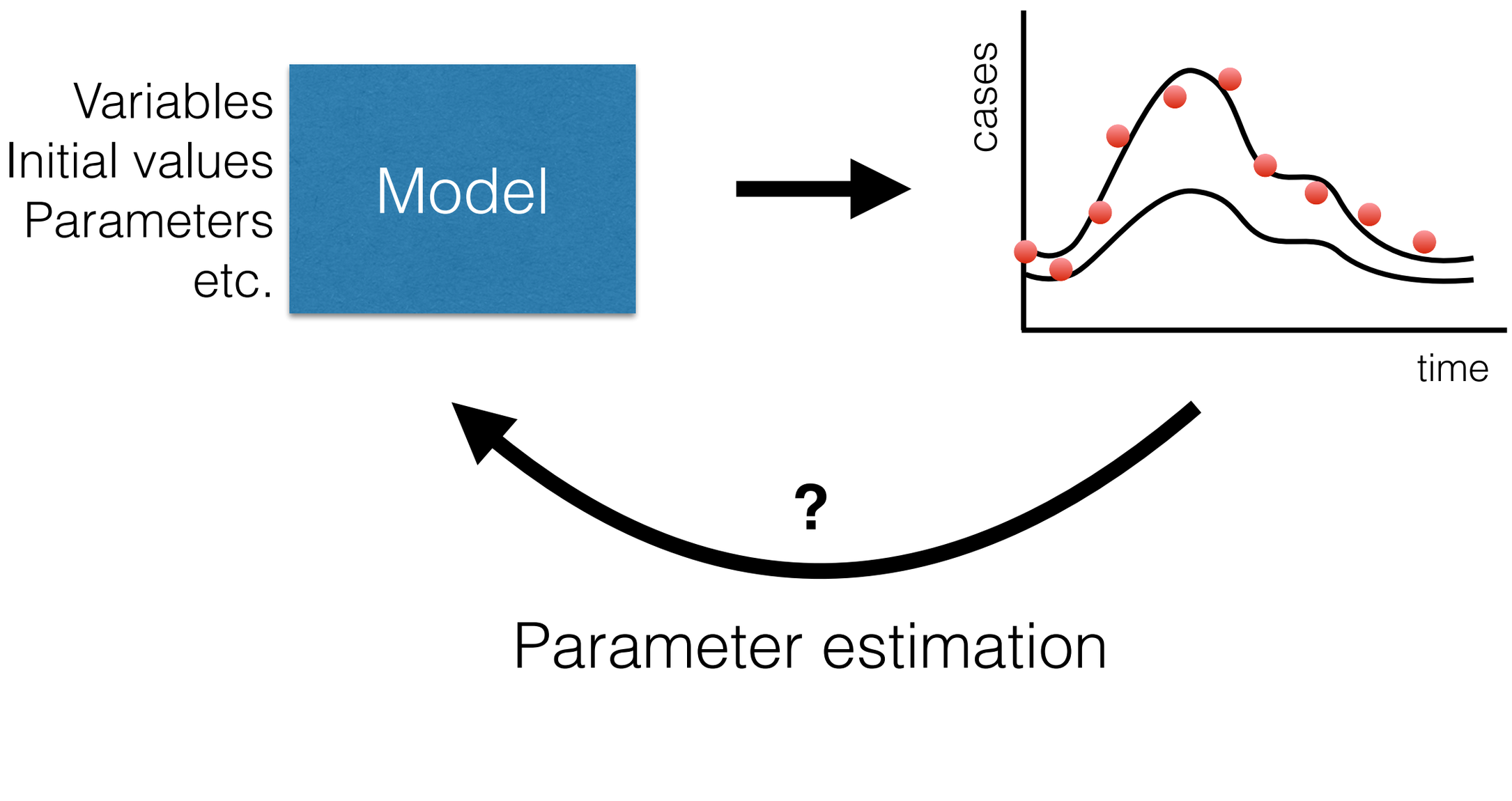
- Depending on parameters, models can give very different results

- How to figure out parameters for model?
    - Direct measurements of parameters often difficult

- If we have data on what is observed in the real world, this may be able to tell us something about what parts of parameter space are more realistic?

- How to connect models with data?

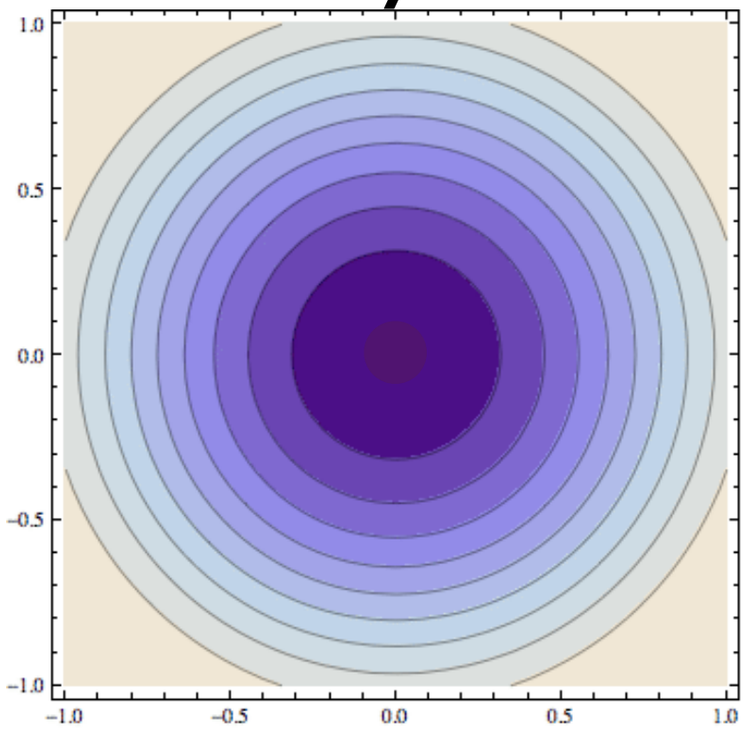
General idea



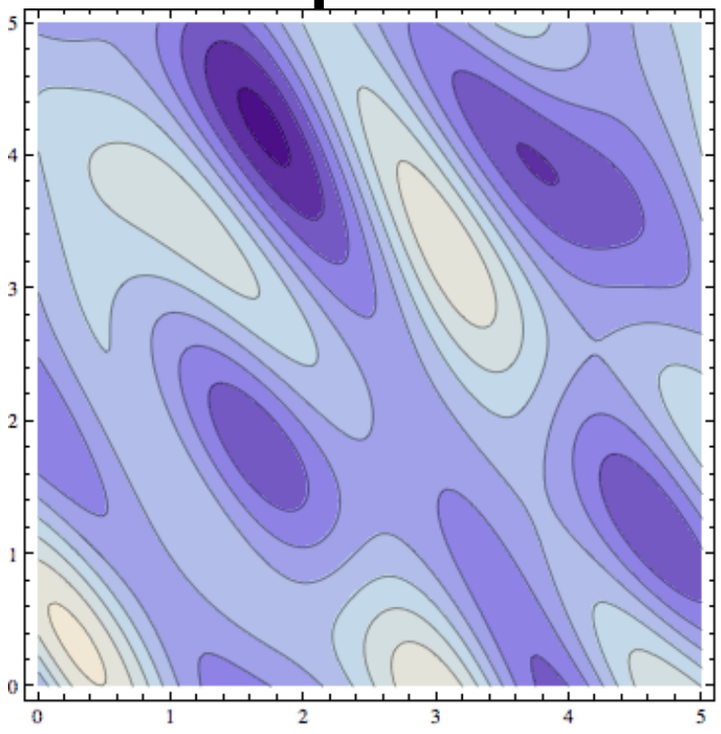
Parameter estimation goals

- In general—search parameter space to find optimal fit to data
- Or to characterize distribution of parameters that matches data

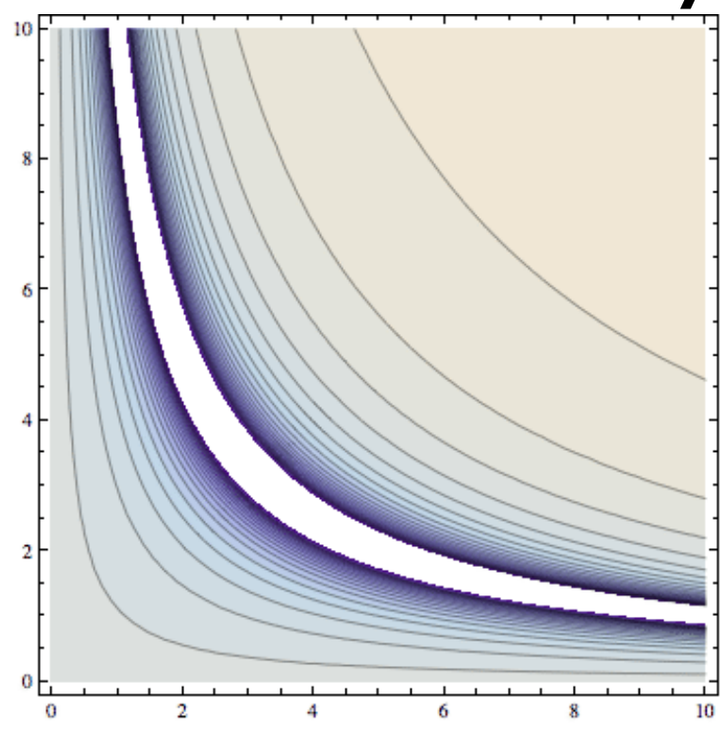
Yay!



Multiple Mins



Unidentifiability



However

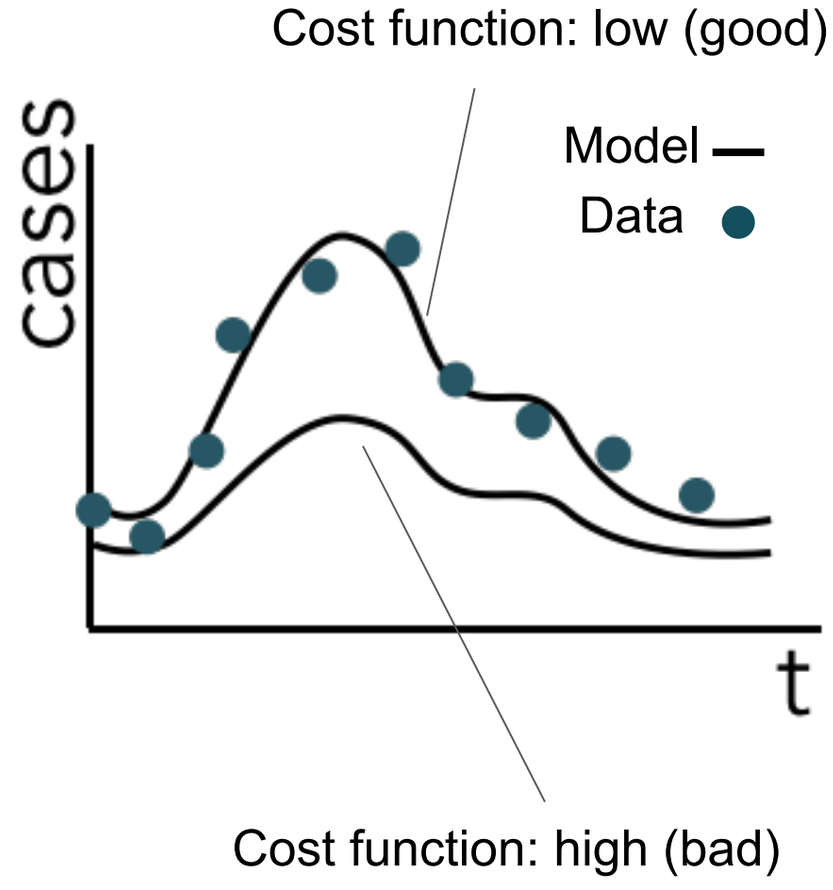
- Parameter estimation is one way to connect models with data—not the only one!
- **Just because a model does not precisely fit the data quantitatively, does not mean it cannot bring useful insight!**
- Usefulness of models is not just about prediction or data fitting
- Qualitative patterns are important
- Often may not have every detail of the mechanism, or may not have enough data to characterize fully, but models can be useful to reason and get intuition about the system—often moreso than a model that ‘fits’ the data better (e.g. think about a mechanistic model vs a spline)

Parameter Estimation

- **Basic idea/assumption**: parameters that give model behavior that more closely matches data are ‘best’ or ‘most likely’

- How to find the ‘closest’ or ‘best’ match?
  - **Cost or objective function**: a way to say how close the model is to the data
  - **Optimization**: a way to adjust the model to get the (a?) best match, i.e. to minimize the cost function

- We want to frame this idea from a statistical perspective (inference, regression)
  - Can determine ‘most likely’ parameters or distribution, confidence intervals, etc.



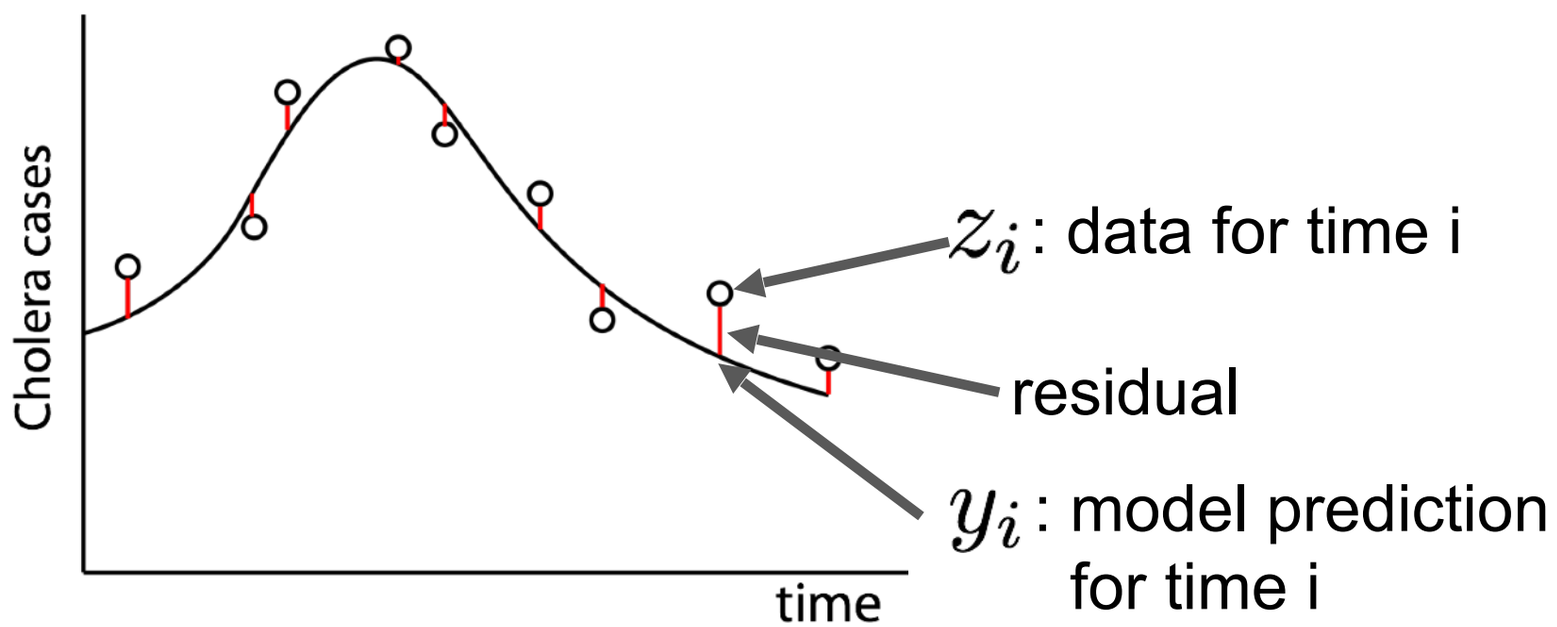
Many things can go wrong!

- Data issues - bias, noise, missing data, not enough data
- Model issues
  - Model misspecification
  - Unidentifiability—particularly for complex models like ABMs, we can expect that many different parameter sets will fit the data equally well

Least squares - one of the most commonly used forms of parameter estimation and optimization

- Goal: adjust parameters to match the data as closely as possible

- Residual: distance between model and data at a given time point



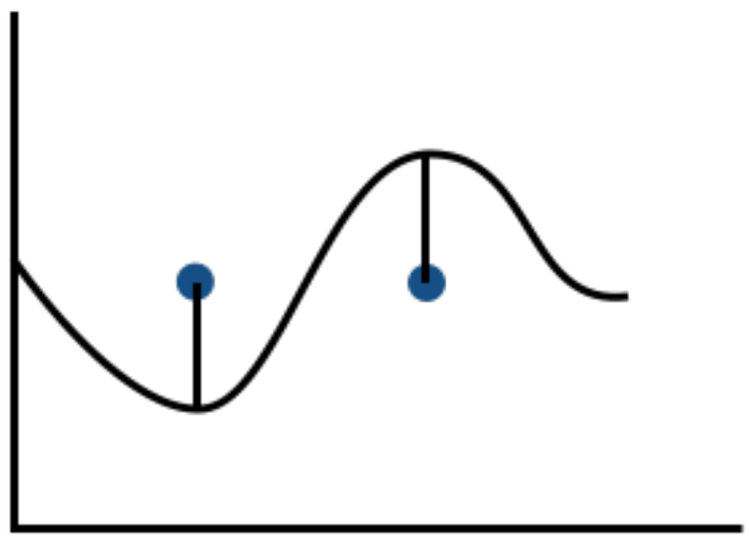
Least squares estimation

- Minimize residuals? Problem is, sign issues

$$\min\left(\sum_{i=1}^{n} z_i - y_i\right)$$

z_i : data for time i

y_i : model prediction for time i



Sum of residuals:
+1 - 1 = 0
even though model is far from data

Least squares estimation

- Least Squares: Minimize square of residuals

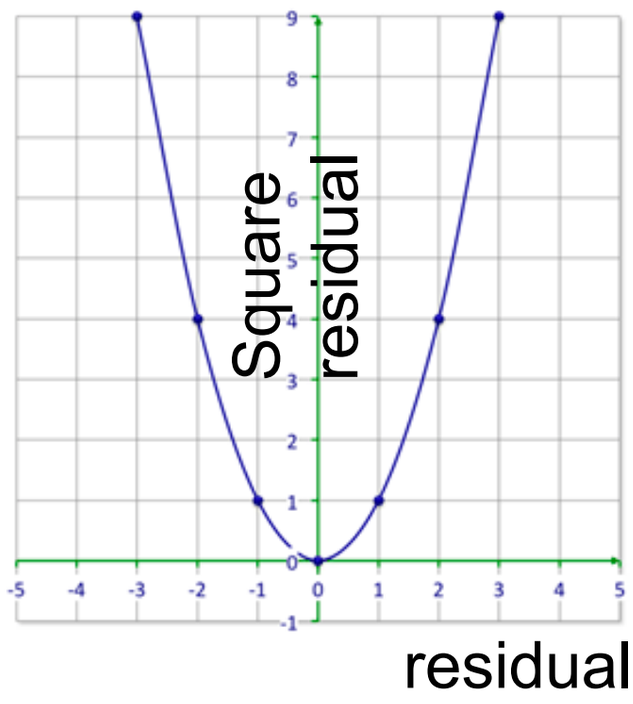
$$\min\left(\sum_{i=1}^{n}(z_i - y_i))^2\right)$$

z_i : data for time i

y_i : model prediction for time i

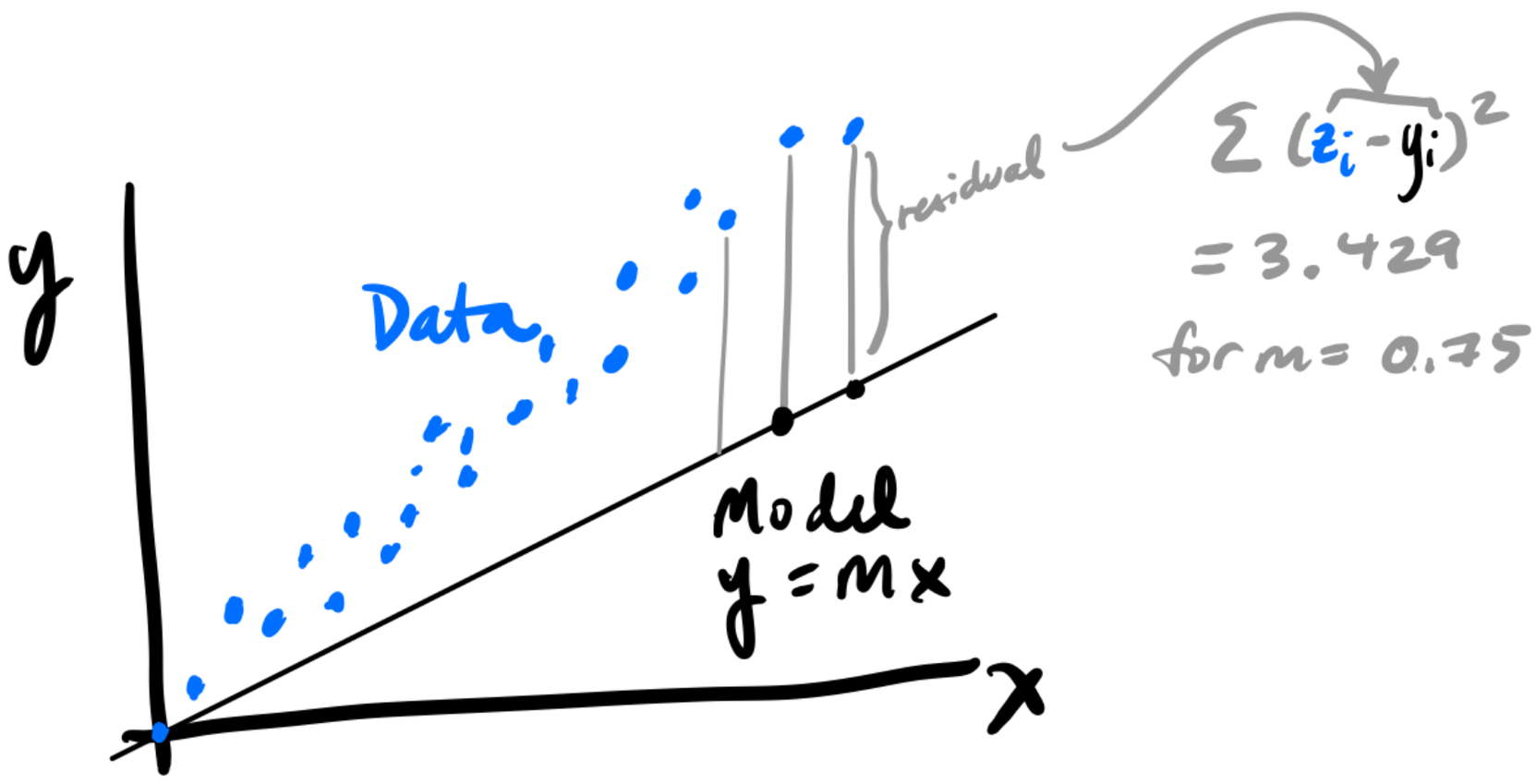
- All errors have same sign in summation

- Penalizes large errors—square term is less than one for small residuals (|zi-yi| < 1) but bigger than 1 for big residuals (|zi-yi| > 1)



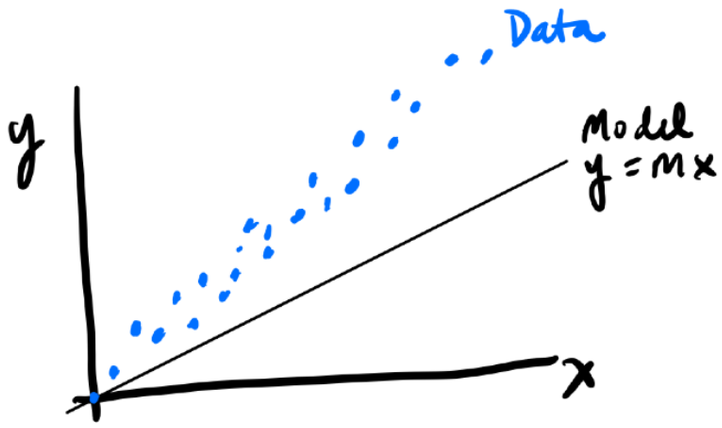
Mini Example

- Model: y = mx
  - 1 parameter, m
- Cost function f(m)
  - Sum of squares
- Initial guess:
  m = 0.75
- Cost function: 3.429

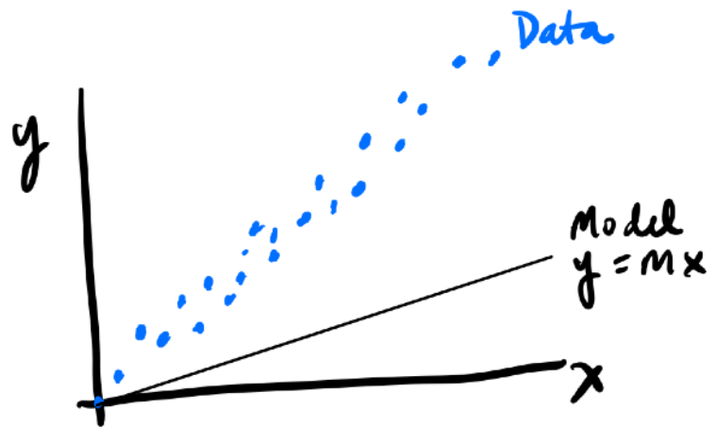


Start

1 m = 0.75, Cost = 3.429

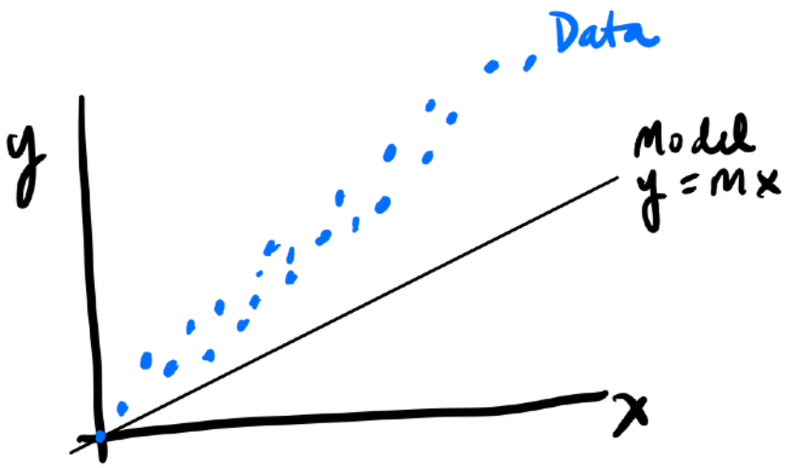


2 m = 0.5, Cost = 3.708

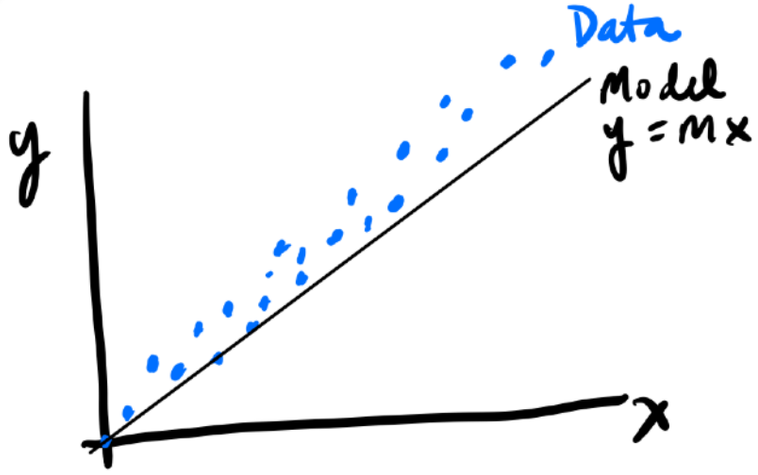


Worse! Let's go back

3 m = 0.75, Cost = 3.429

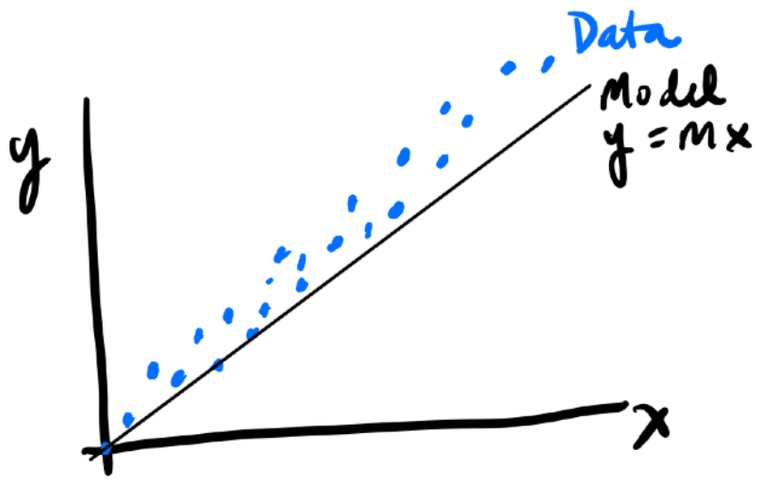


4 m = 1, Cost = 1.654

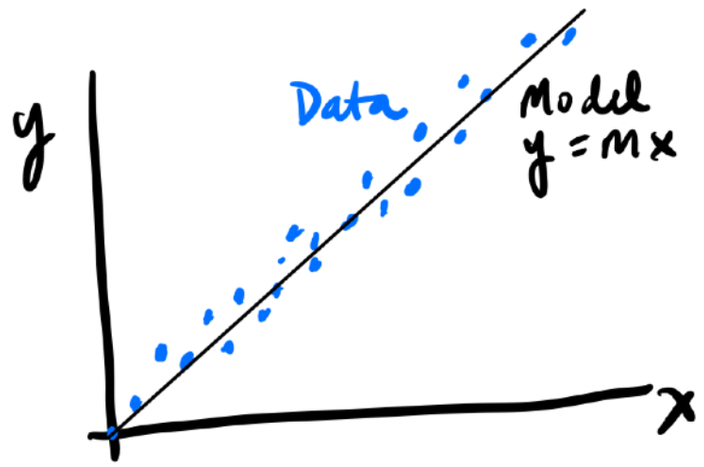


Increasing m is better

4 m = 1, Cost = 1.654



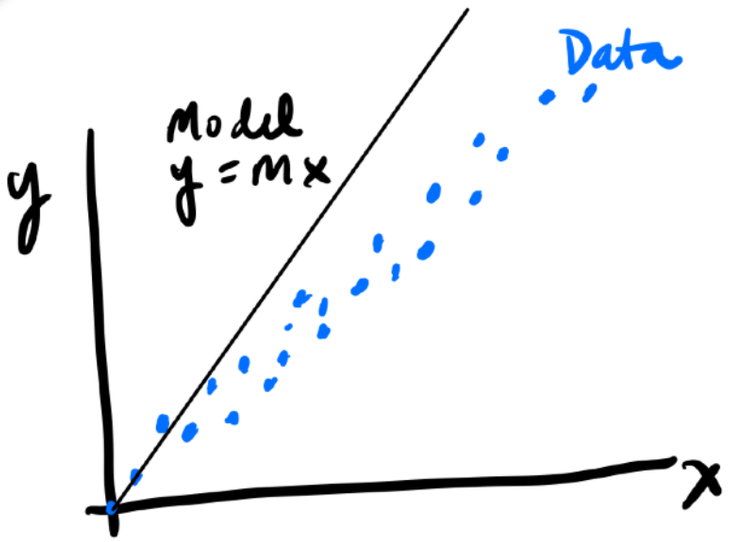
5 m = 1.2, Cost = 0.0016



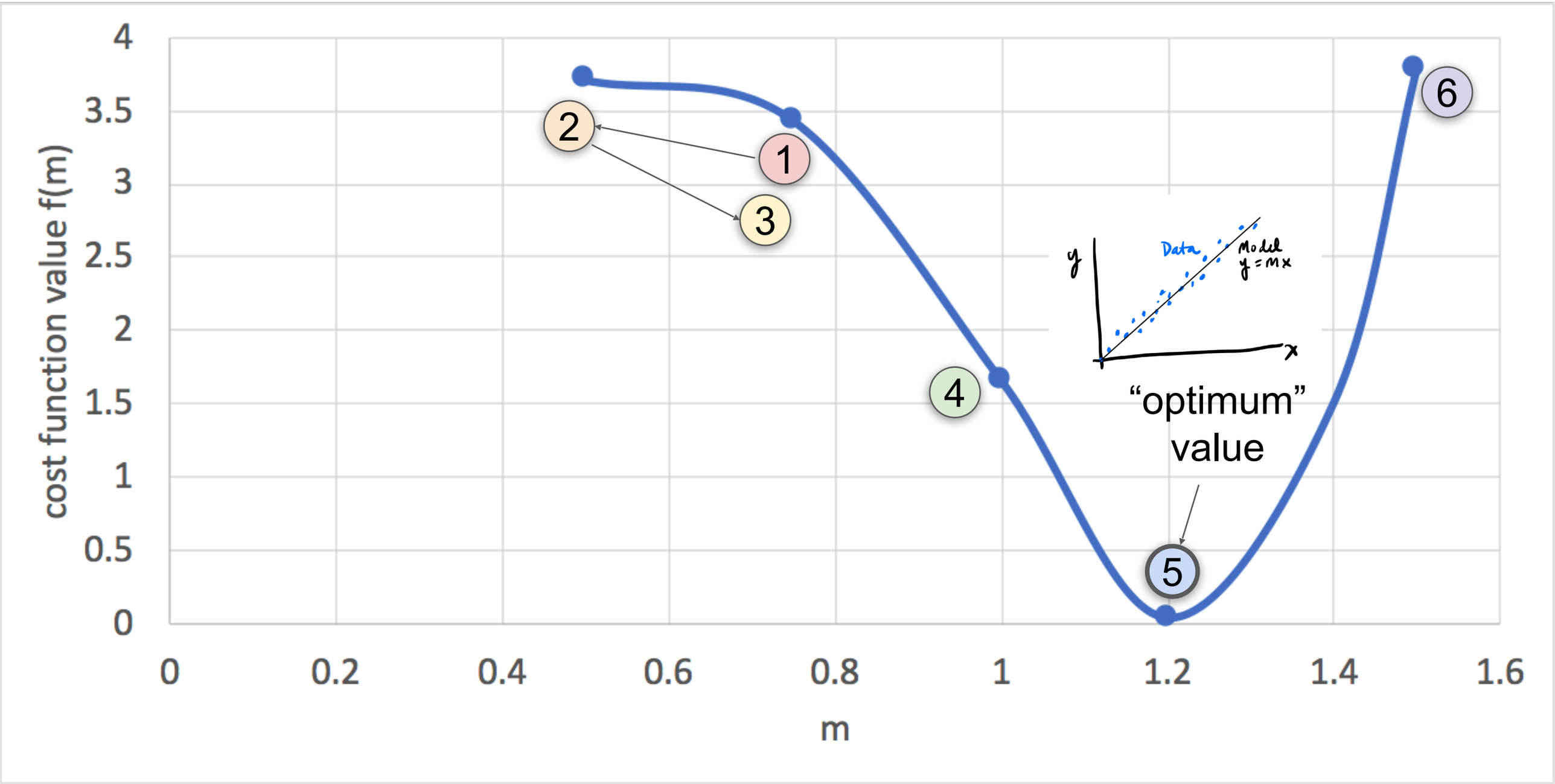
This looks pretty good! Let's try a little further

6 m = 1.5, Cost = 3.781

Went too far!



Okay, so m = 1.2 seems the best of the guesses we've tried! Let's plot our progress


cost function value f(m)
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
m
1
2
3
4
5
6
y
x
Data
Model
y = mx
"optimum"
value

Optimization

Basic idea: use mathematical and computational methods to make something optimal (best)

In our case, usually what we mean is to choose a function or formula that represents what we want to optimize (e.g. how close or far is our model from the data, or how well a control strategy is working, or the actual cost of something), and find the minimum or maximum value!

This function is called our **cost or objective function**

Vocabulary

- **Objective function or cost function**, f(x): the function we are trying to minimize or maximize (e.g. goodness of fit). Can be a function of a range of different variables!
    - Usually a function of our data and the parameters and variables of our model
    - Historically, convention is to minimize f(x) (note if you want to maximize just minimize -f(x) !)

- **Variables/inputs/parameters**: inputs to the cost function that we can adjust/change/control

- **Constraints**: any restrictions to our optimization (e.g. find the biggest number, but our constraints are that it has to be even and less than 11)

Optimization methods

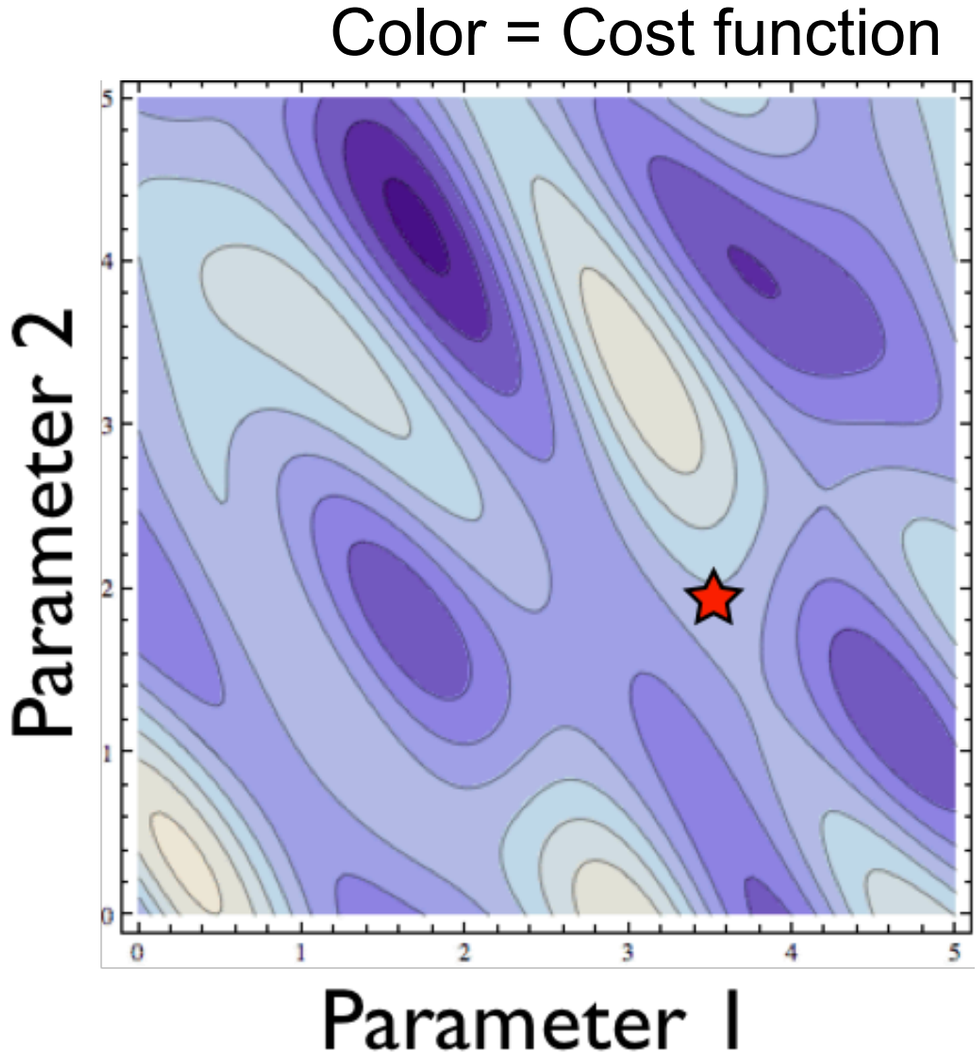
Many methods do something like:

- Take in a set of starting parameter values (initial values)

- Evaluate the nearby cost function landscape (i.e. what are nearby cost function values)

- Pick a new set of parameter values

- Repeat until no further improvement can be made, or enough parameter values have been sampled



Optimization methods

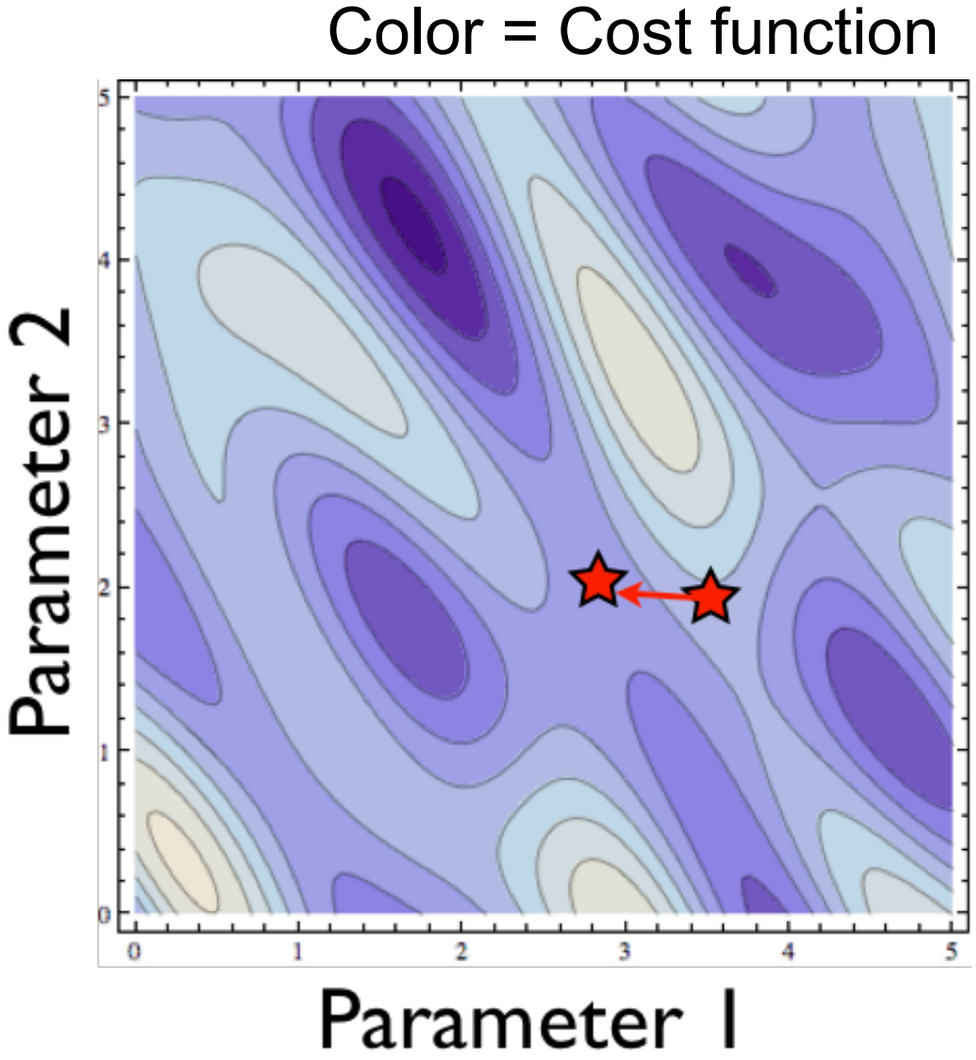
Many methods do something like:

- Take in a set of starting parameter values (initial values)

- Evaluate the nearby cost function landscape (i.e. what are nearby cost function values)

- Pick a new set of parameter values

- Repeat until no further improvement can be made, or enough parameter values have been sampled



Optimization methods

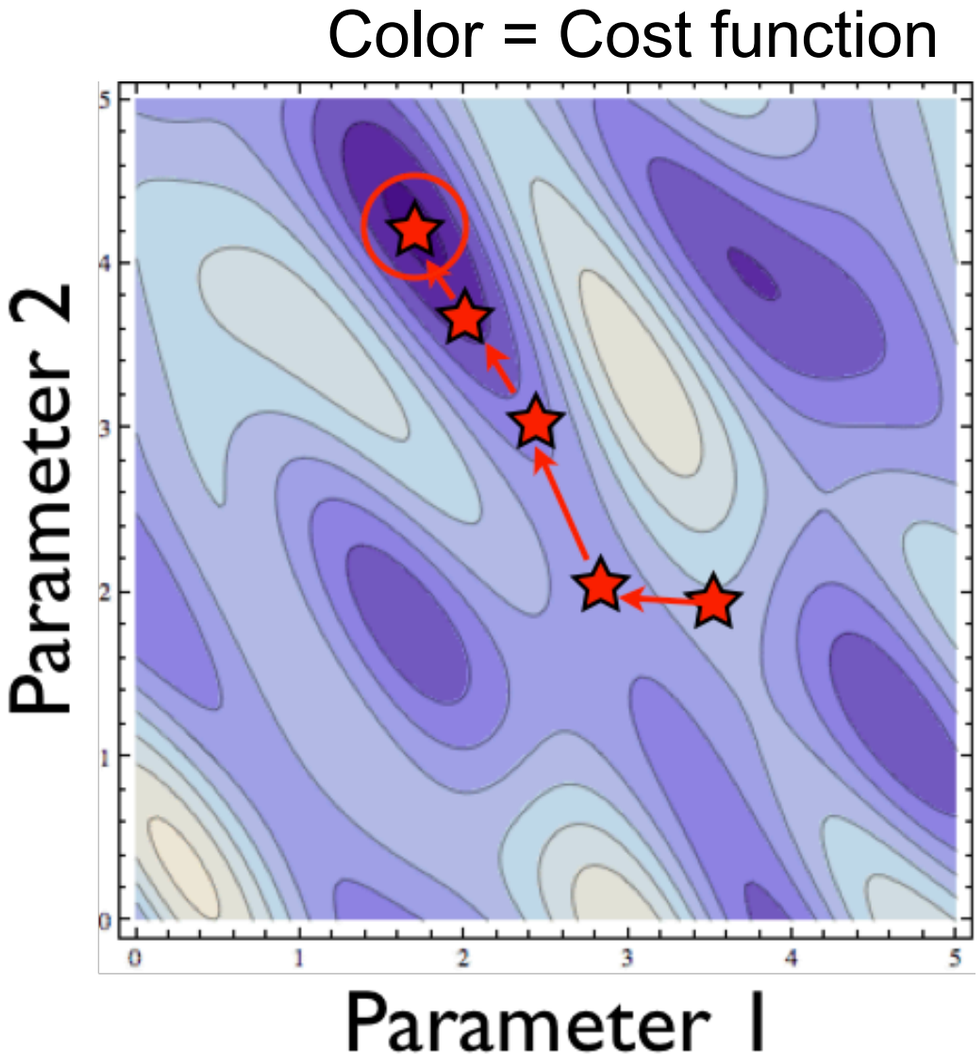
Many methods do something like:

- Take in a set of starting parameter values (initial values)

- Evaluate the nearby cost function landscape (i.e. what are nearby cost function values)

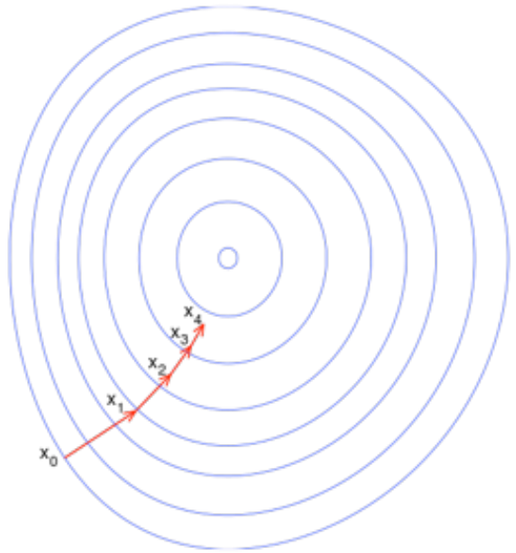
- Pick a new set of parameter values

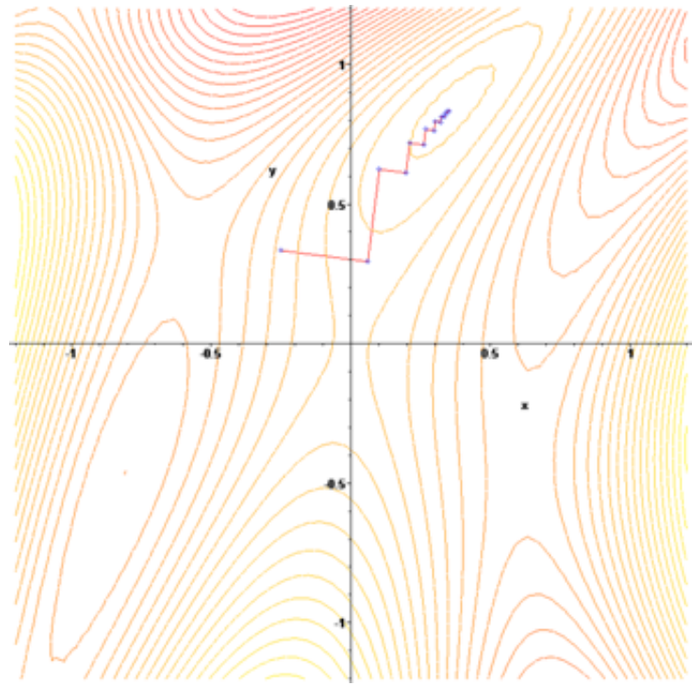
- Repeat until no further improvement can be made, or enough parameter values have been sampled



Optimization methods: gradient descent

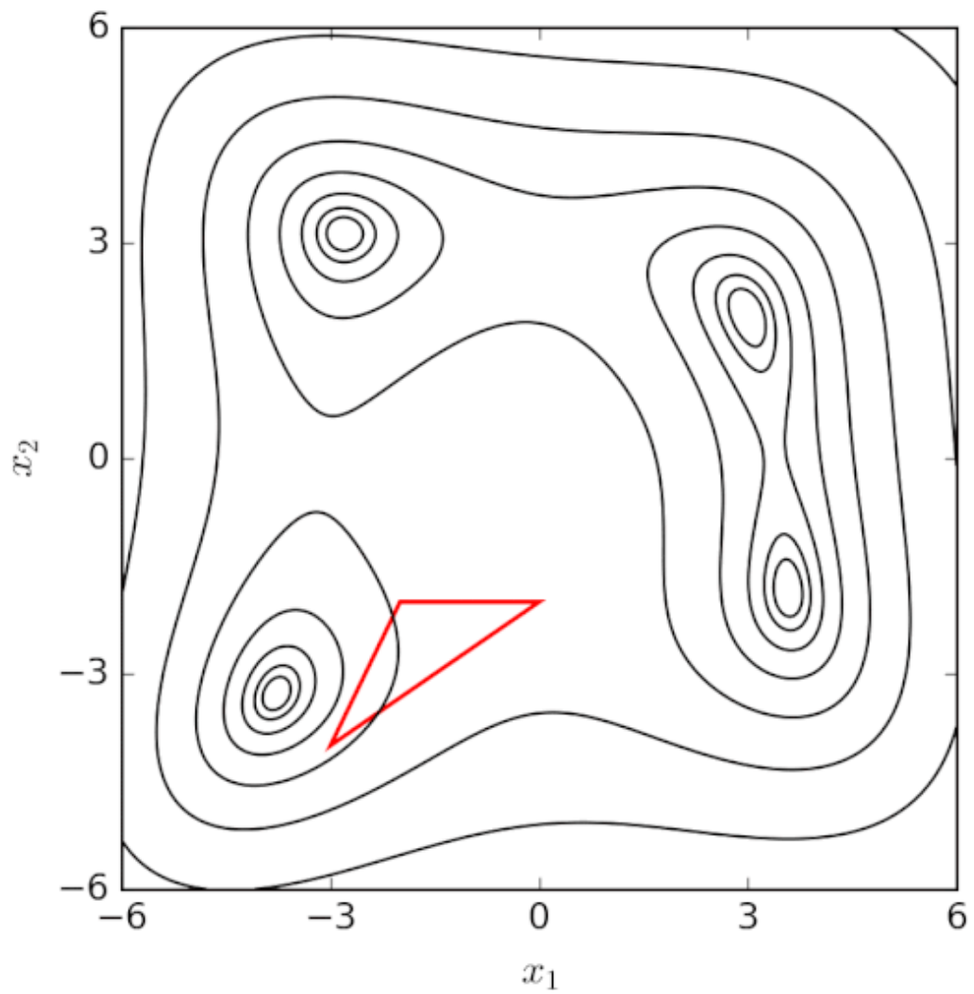
- Fast!
- Relatively easy to use and implement
- Can get stuck in local minima
- Some surfaces can cause weird/problematic behavior (zig-zag issue, soft serve ice cream example)





Optimization methods: simplex algorithms

- Can be similar to gradient search
  - Easy to implement
  - Fast, efficient to compute
- Gets around some of the issues of gradient descent, but can still get caught in local minima
- E.g. Nelder-Mead algorithm



Global optimization methods

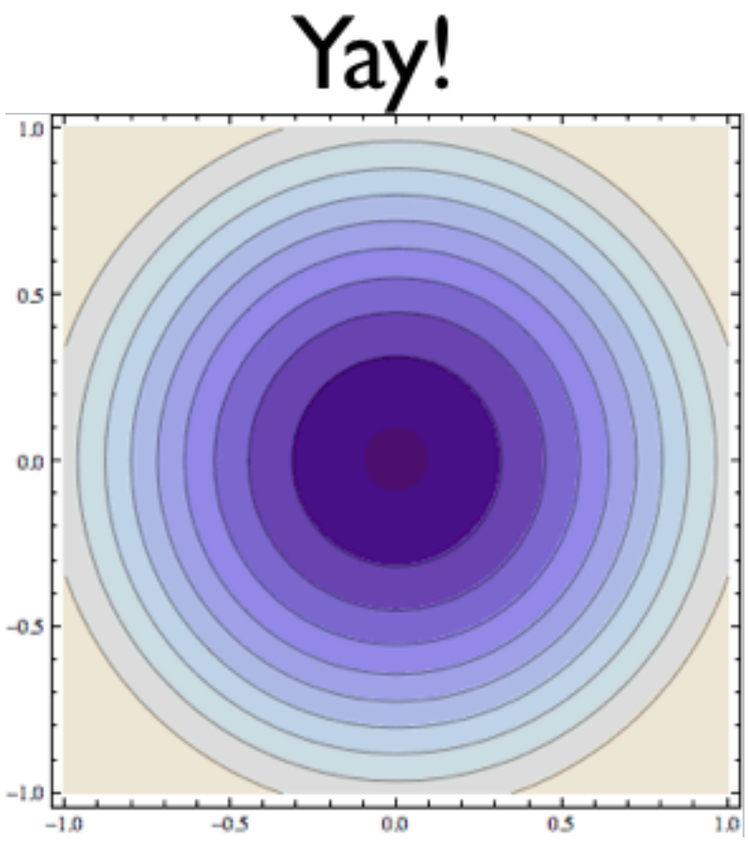
- Many options! Simulated annealing, evolutionary algorithms, etc
    - Markov Chain Monte Carlo (MCMC) methods can also be considered in this category

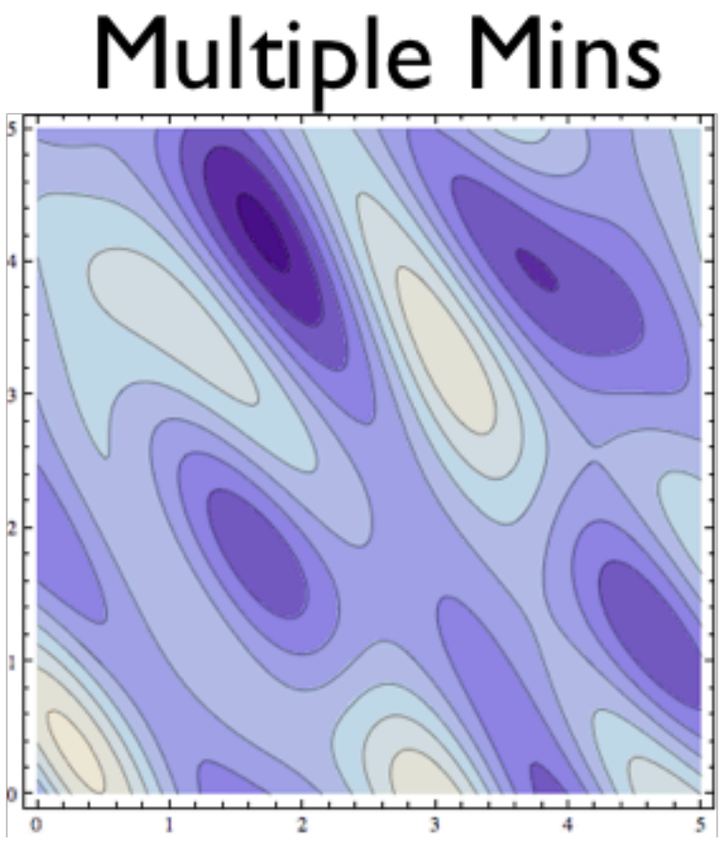
- These approaches typically allow for some acceptance of worse cost function values, to allow the optimizer to escape local minima

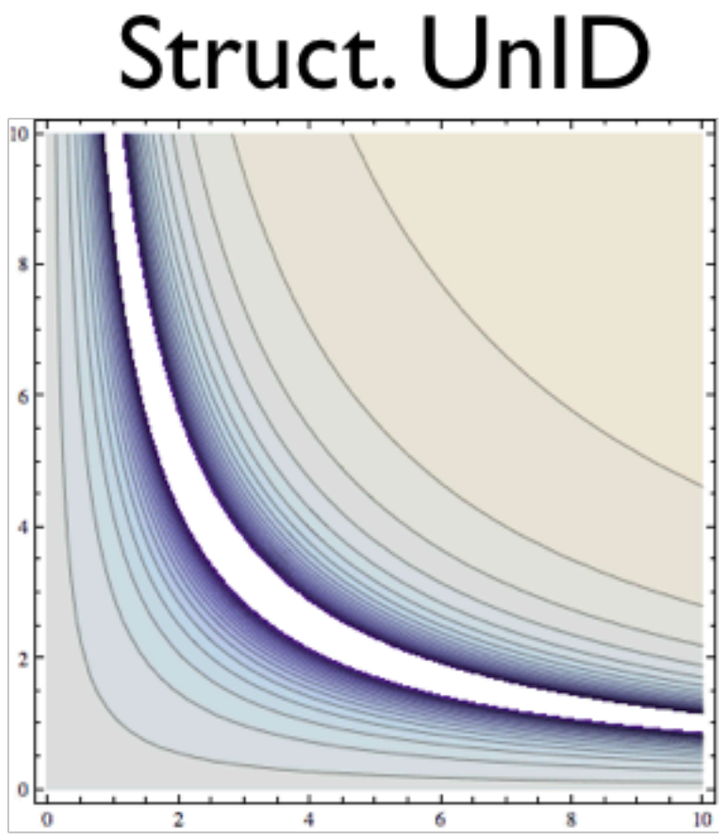
- Also often involve examining many different initial parameter values or many parameter samples, to more fully explore the cost function surface

- Gives a better view of the cost function surface—but often very(!) slow compared to local optimization methods (gradient descent, simplex methods, etc)

Optimization methods are often formulated for convex, single minimum surfaces—multiple minima and canyons (sometimes called unidentifiability) can cause problems







Code examples parts 1 and 2

How to frame this statistically?
Maximum Likelihood

- View our model as a probability distribution, where we suppose we know the general form of the probability density function but not the parameter values

- Then if we knew the parameters we could calculate probability of a particular observation/data:

$$P(z \mid p)$$

data ↗ ↖ parameters

- Figure out which parameter values make the model most likely to generate the data we see

Maximum Likelihood

Given the parameters, we can usually work out the probability of observing a particular data set $P(z \mid p)$

For example, suppose we have a potentially biased coin, and we want to estimate the probability that a coin flip results in heads—call this parameter p.

- Our data (call this z): we measure 3 coin flips: H, T, H.

- What's the probability that p = 0.5 given this data? Hard to say!

- But, if we knew p, we can definitely calculate the probability of seeing this data. Assume the coin flips are independent, then:

  P(z|p) = p*(1-p)*p

This P(z|p) is the likelihood! The main idea of maximum likelihood is to choose p to maximize this

Maximum likelihood estimate?

Data (z) is 3 coin flips: H, T, H

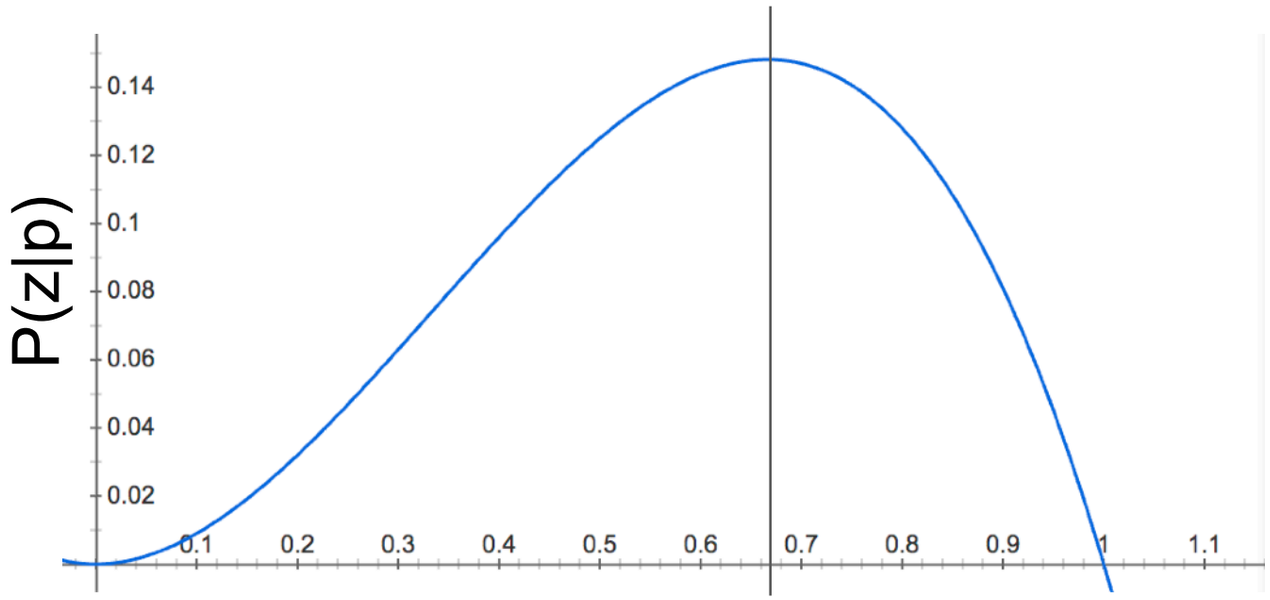
Likelihood function:
$P(z|p) = p*(1-p)*p$

What value of p maximizes this?

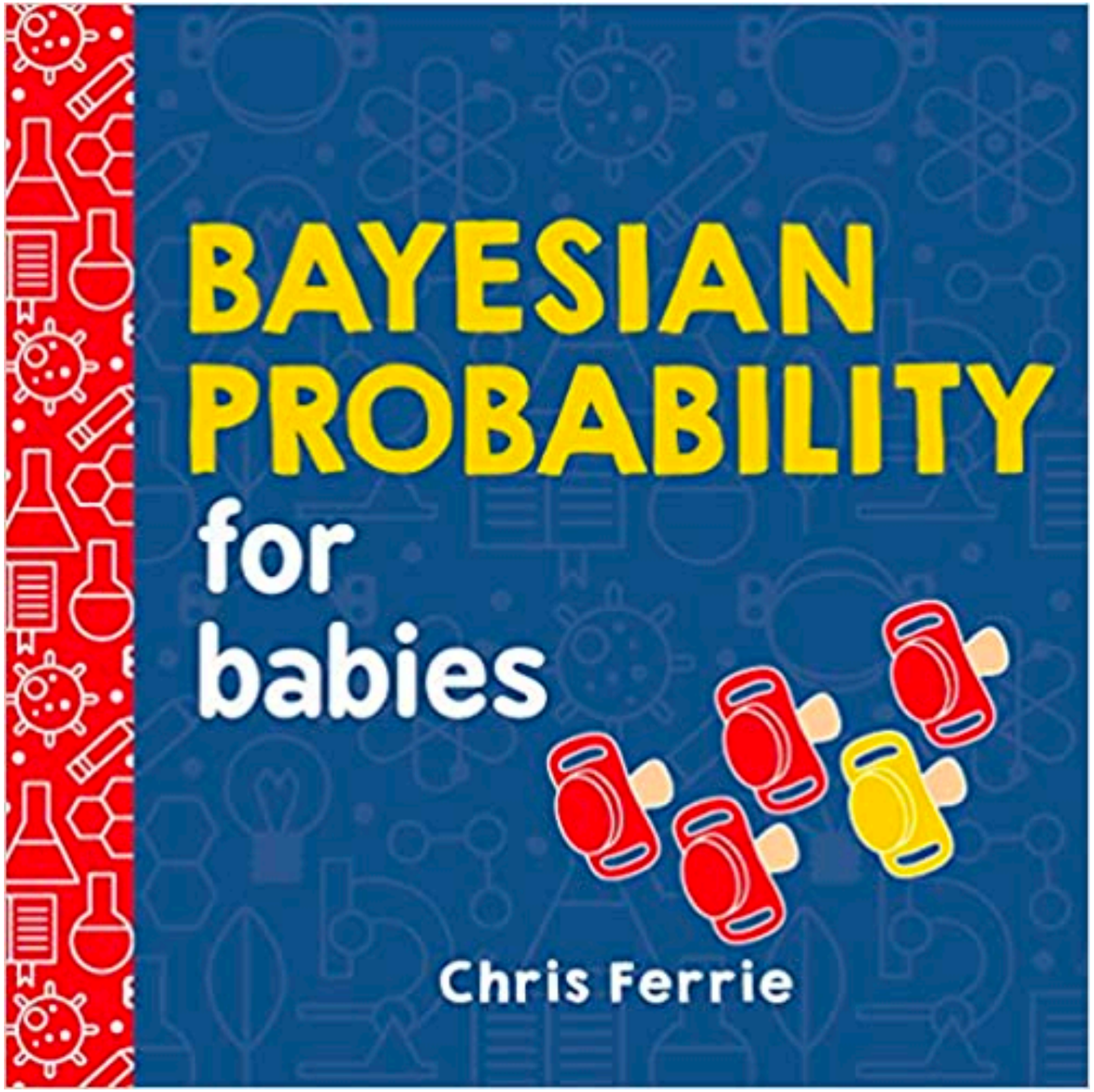
$P(z|p) = p*(1-p)*p = p^2 - p^3$

(Note this is actually a constrained optimization problem—p must be between 0 and 1)

We could code it up—but we can also just plot it!



Maximum likelihood estimate!
p = 2/3


BAYESIAN
PROBABILITY
for
babies
Chris Ferrie

Some cookies have candy.

Some don't.

Our data:

Our model:

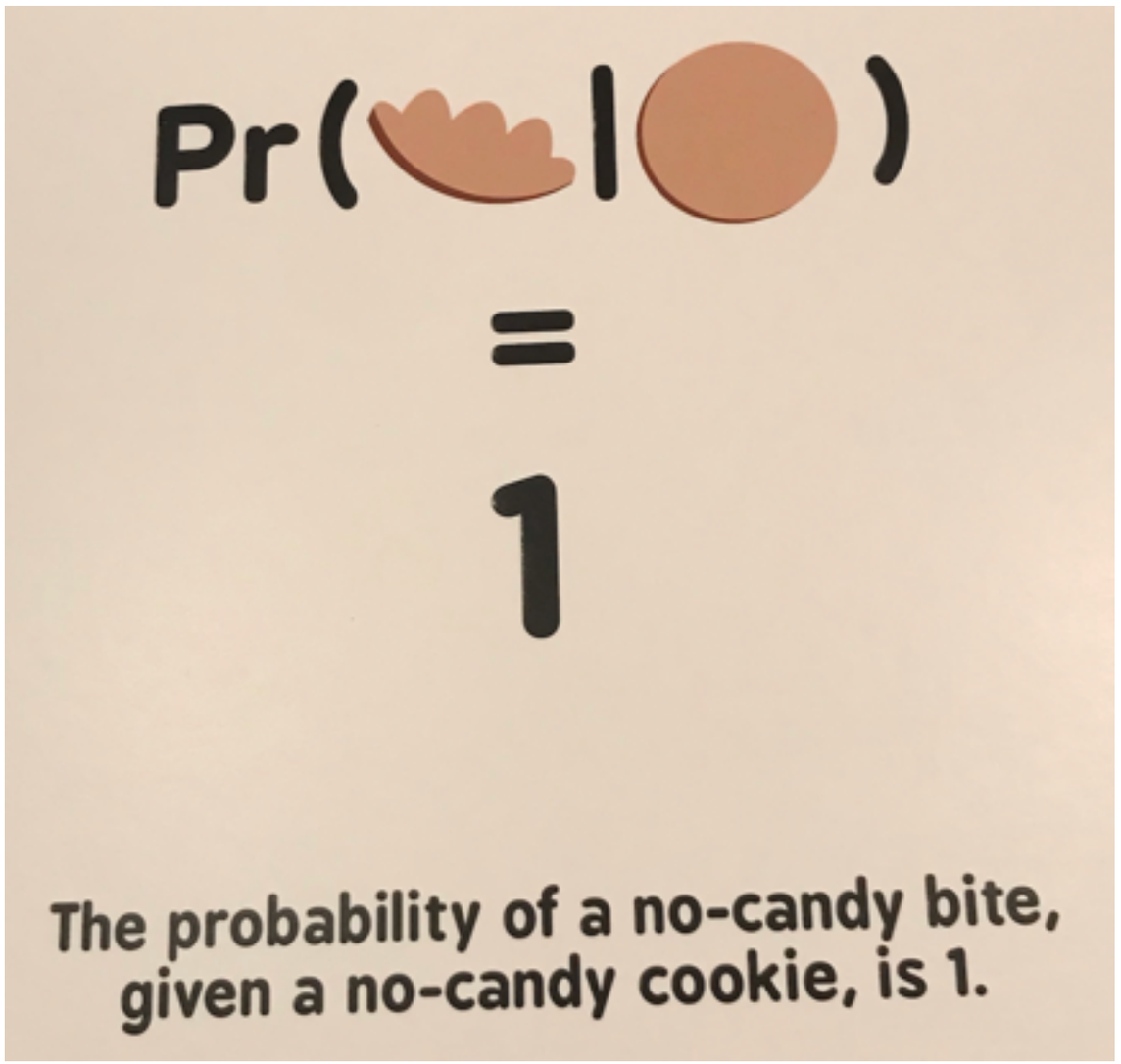
- Candy cookie vs. no-candy cookie
- Our model parameters? Just one—an indicator variable c, where:
  - c = 0 if it was a no-candy cookie
  - c = 1 if it was a candy cookie

In other words, our parameter estimation problem is to estimate c!

Does c = 0 or 1?

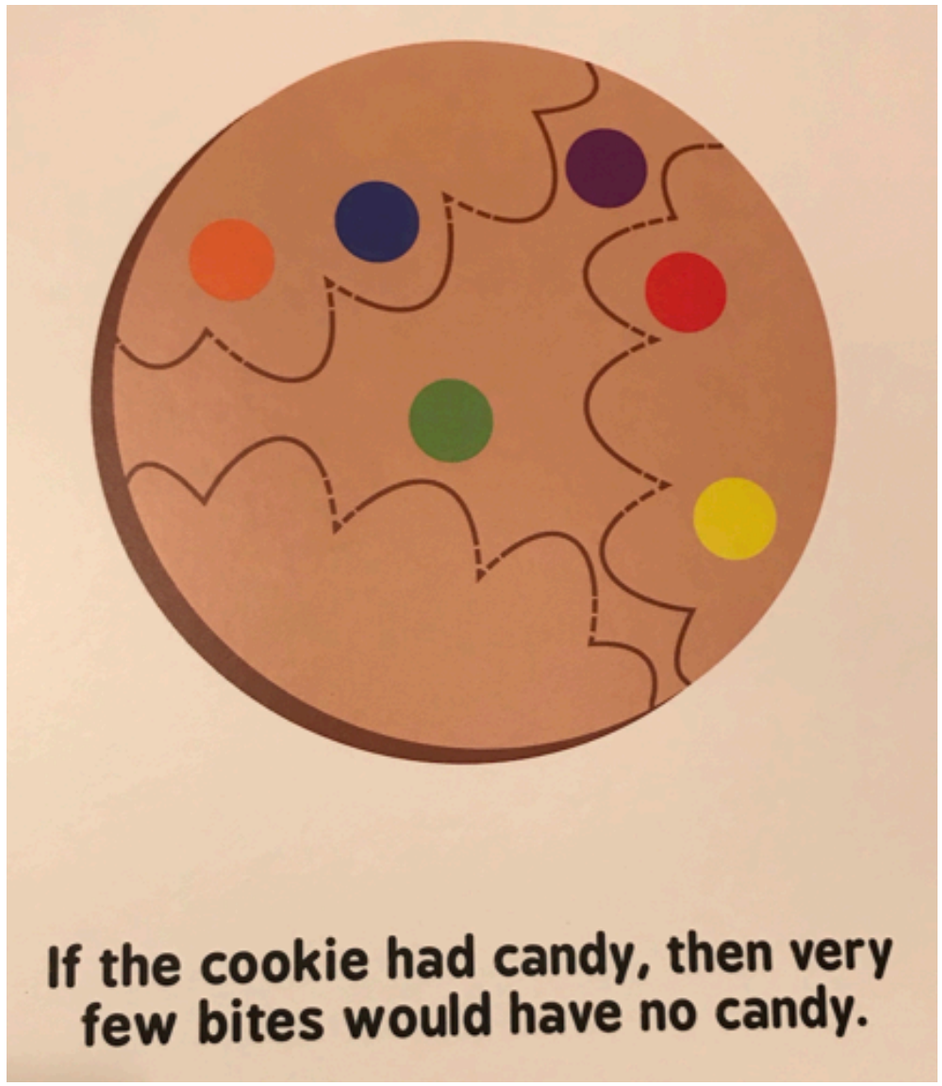
What is the likelihood?

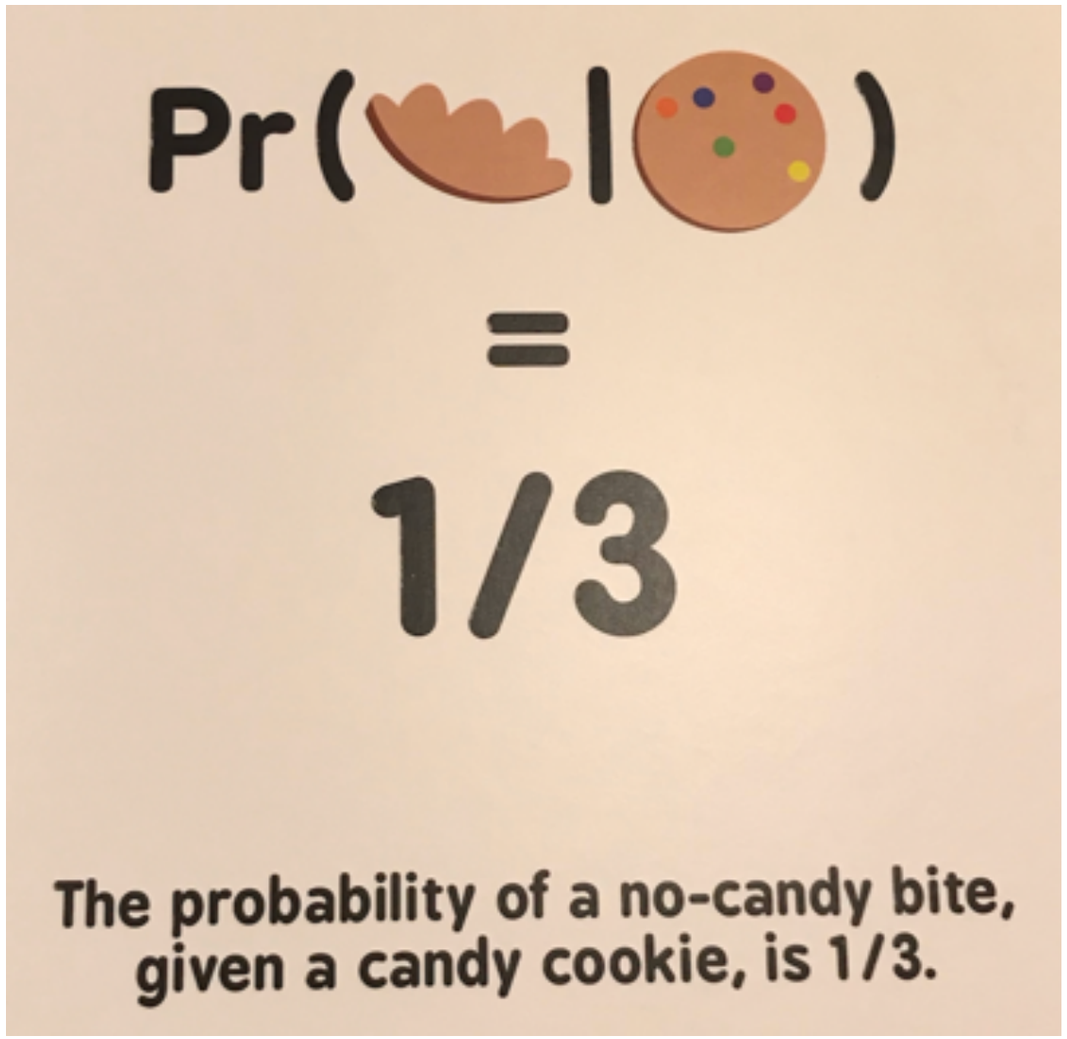
$$L(c = 0 \mid \text{NC bite}) = P(\text{NC bite} \mid c = 0) = 1$$



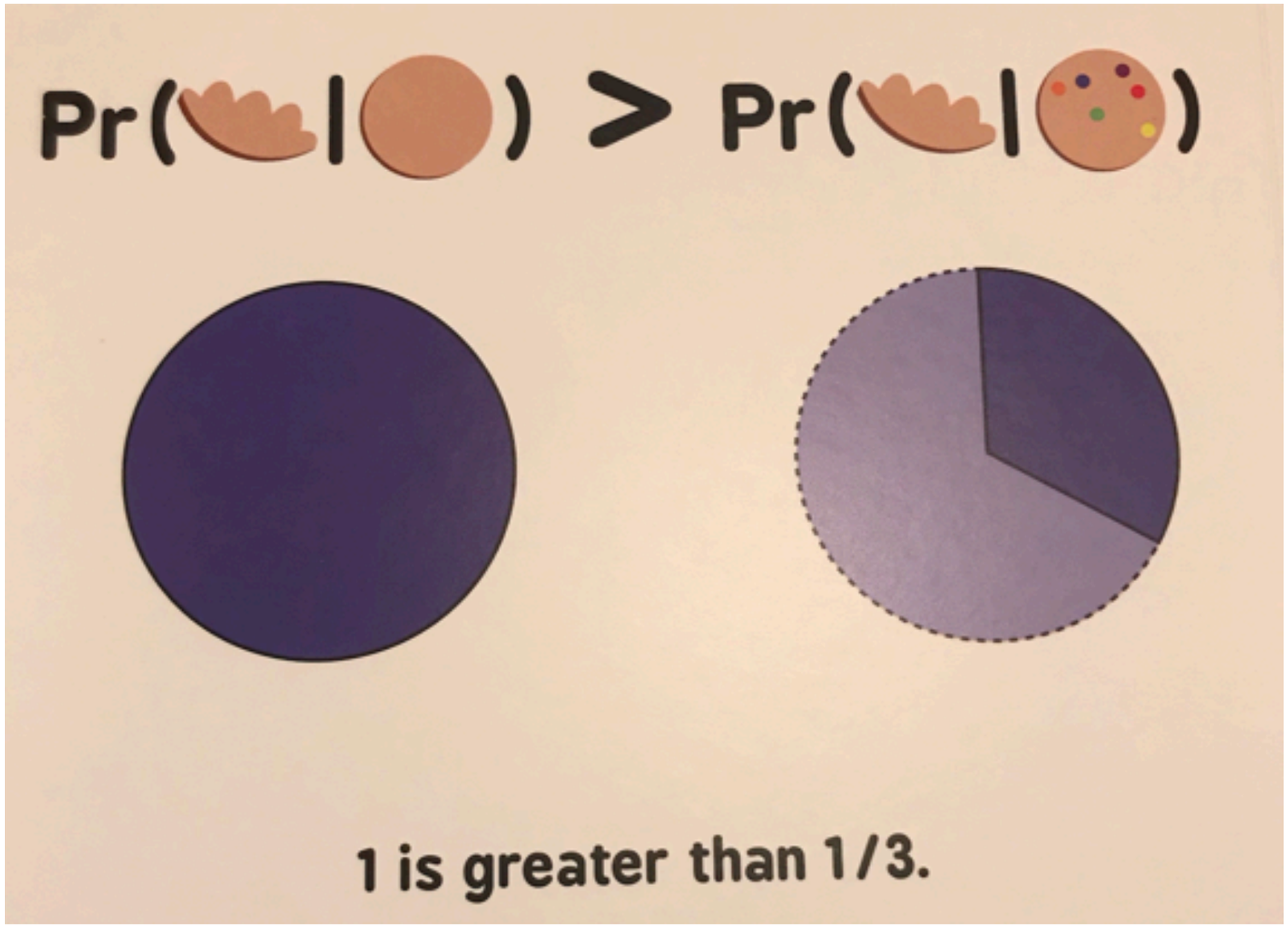
What is the likelihood?

$$L(c = 1 \mid \text{NC bite}) = P(\text{NC bite} \mid c = 1) = 1/3$$





What is the maximum likelihood estimate?
1 > ⅓, so we estimate that c = 0



Maximum Likelihood

- **Likelihood Function**: the probability of seeing the data we have, assuming that we knew the parameter values

$$P(z \mid p) = f(z, p) = L(p \mid z)$$

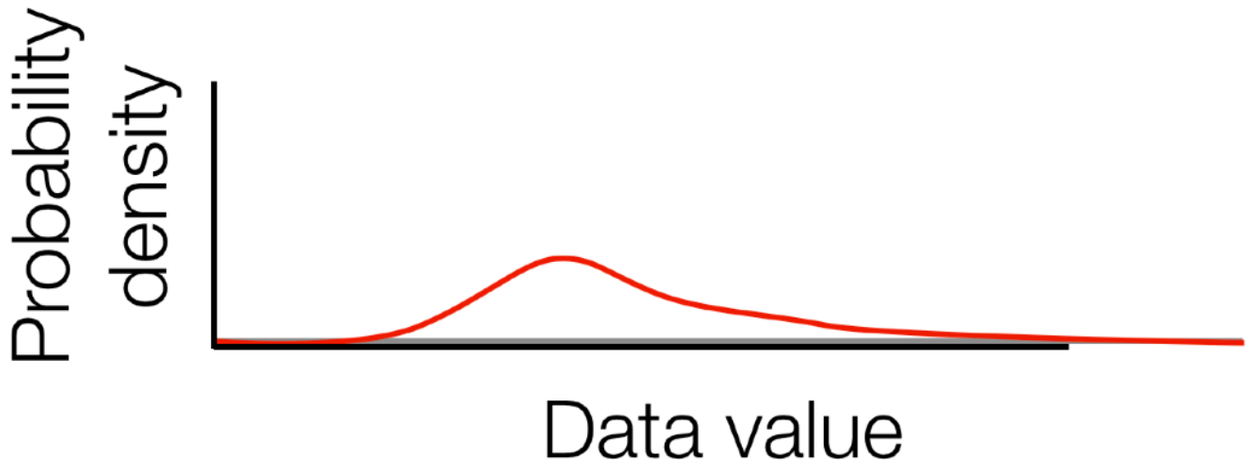
- Re-think the distribution as a function of the data instead of the parameters

- E.g. $f(z \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(z-\mu)^2}{2\sigma^2} \right] = L(\mu, \sigma^2 \mid z)$

- Find the value of p that maximizes L(p|z) - this is the maximum likelihood estimate (**MLE**) (most likely given the data)—in other words, L becomes our cost function! (usually actually - log(L) but this is the idea!)

Likelihood function

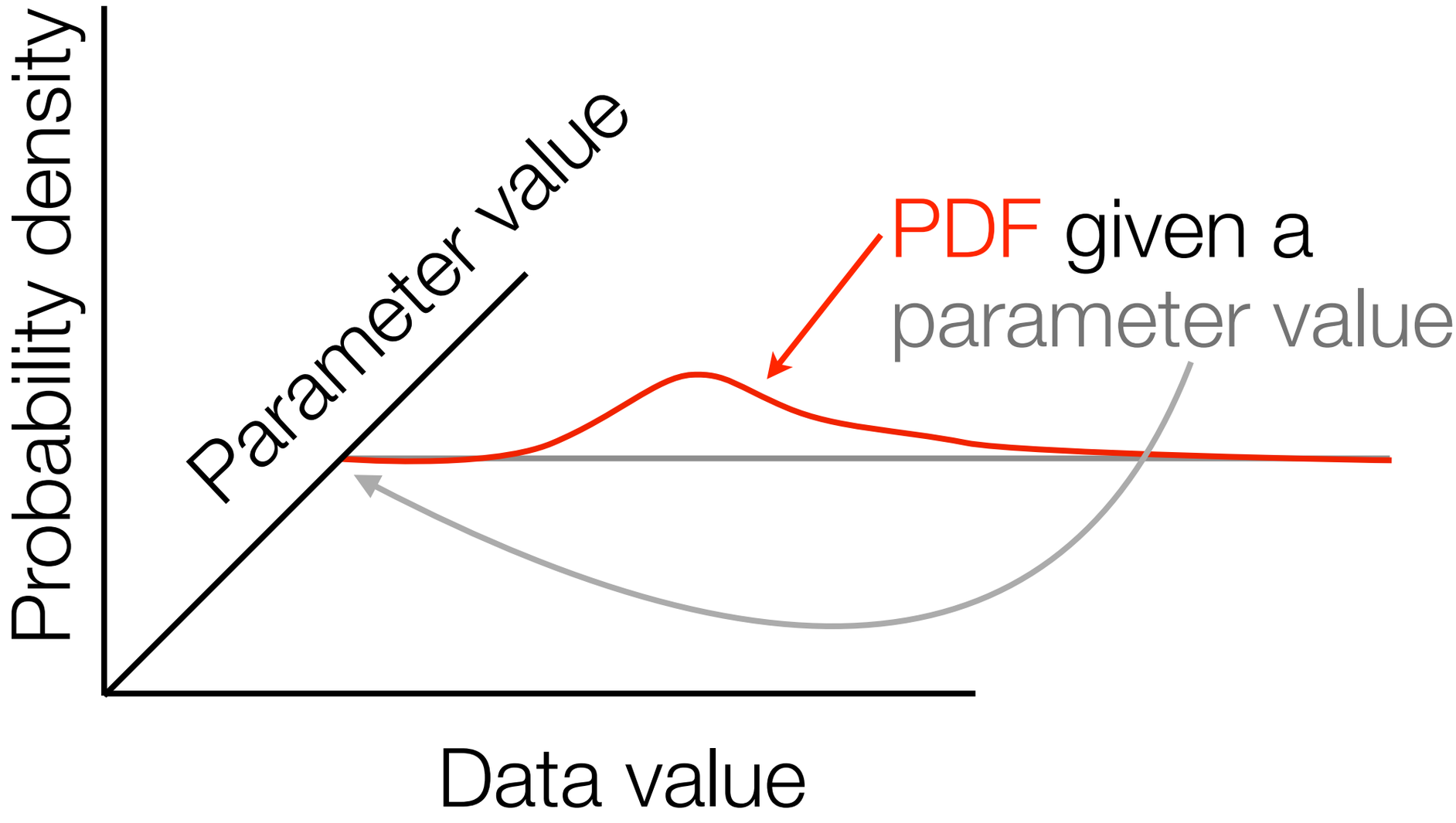
We usually plot a probability distribution something like this:



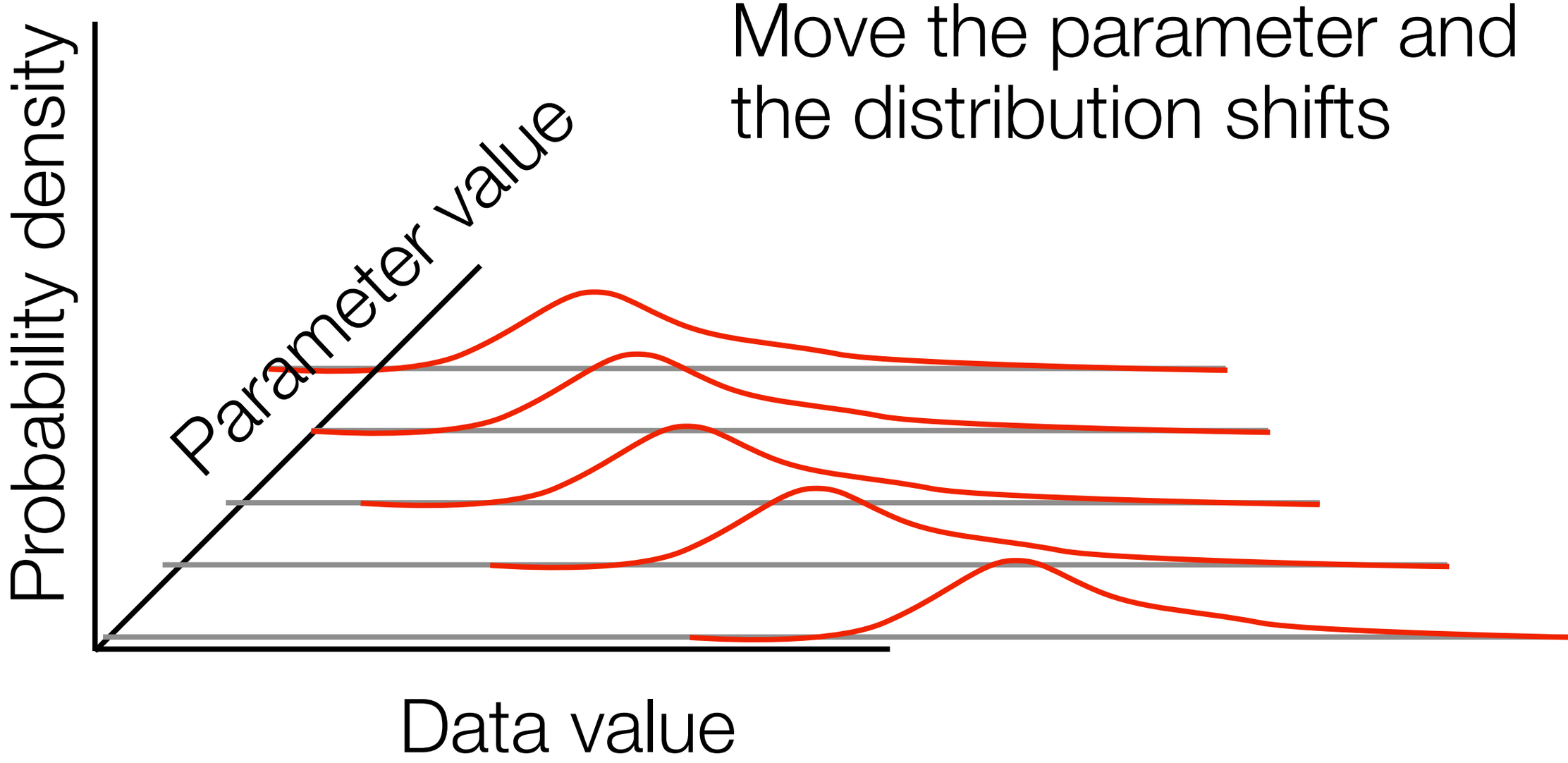
But, there are parameters that control the shape of this distribution! The mean, standard deviation, etc.

Really, we should maybe think of it more like:

Likelihood Function

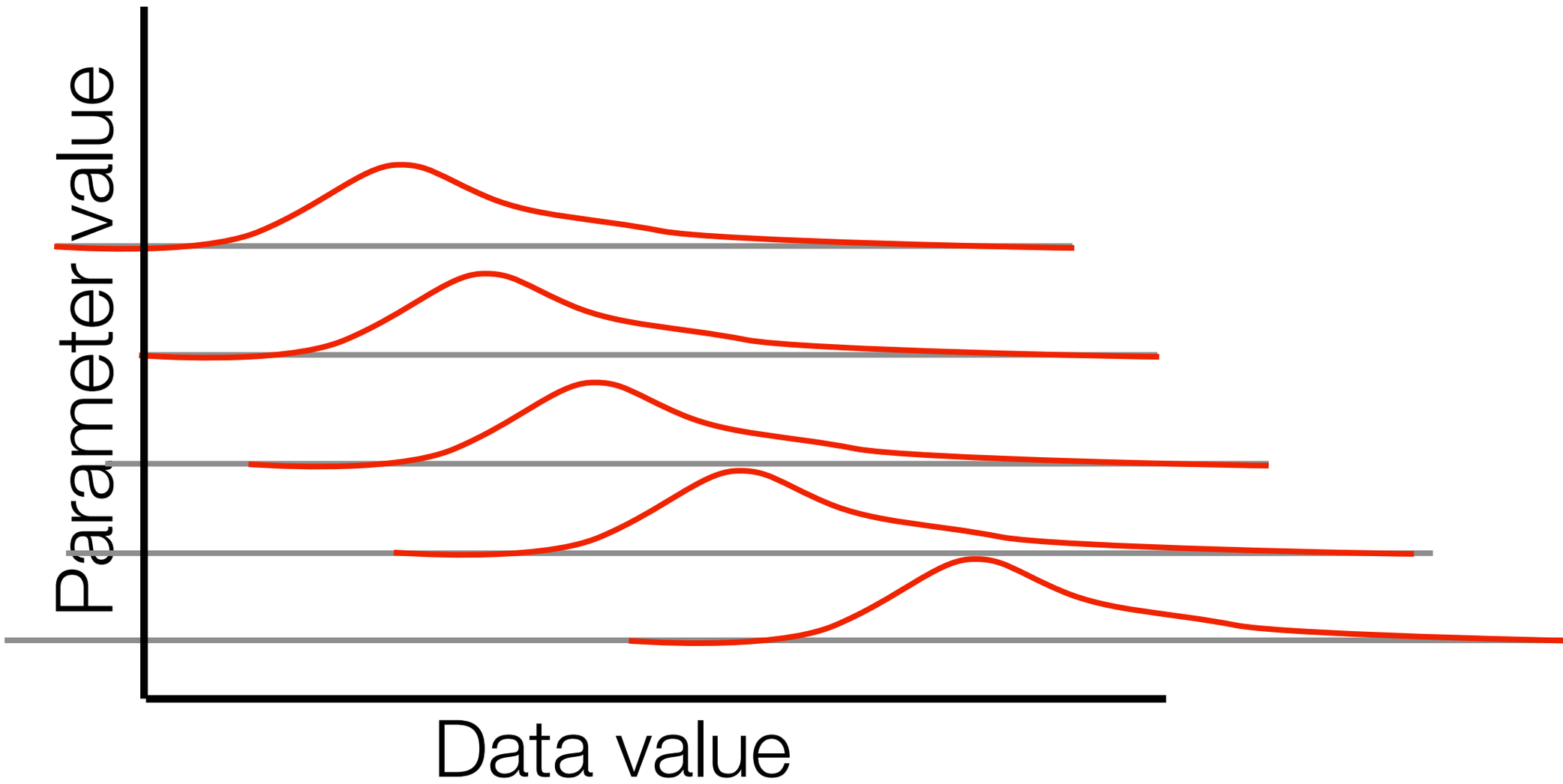


Likelihood Function

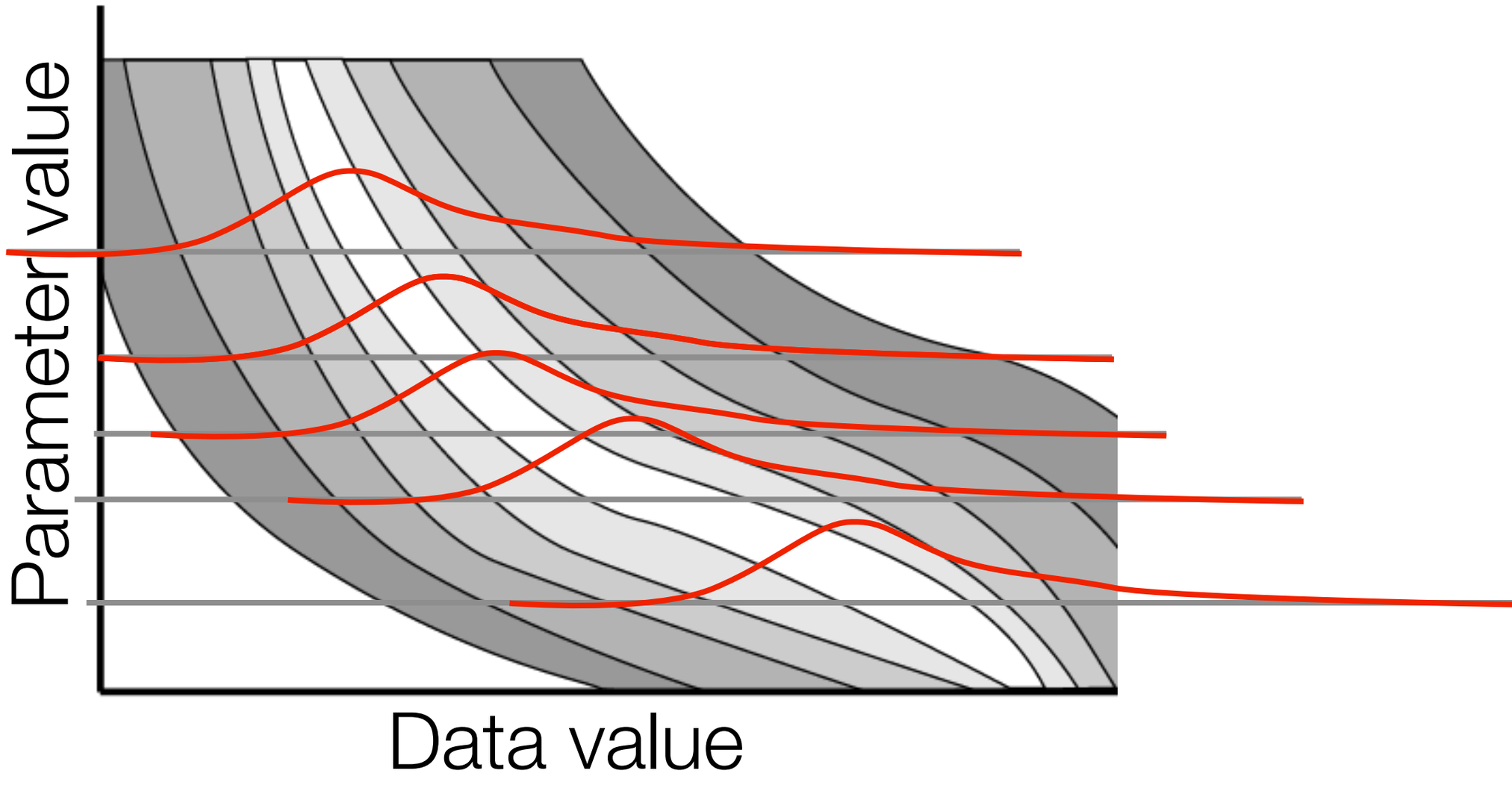


Move the parameter and the distribution shifts

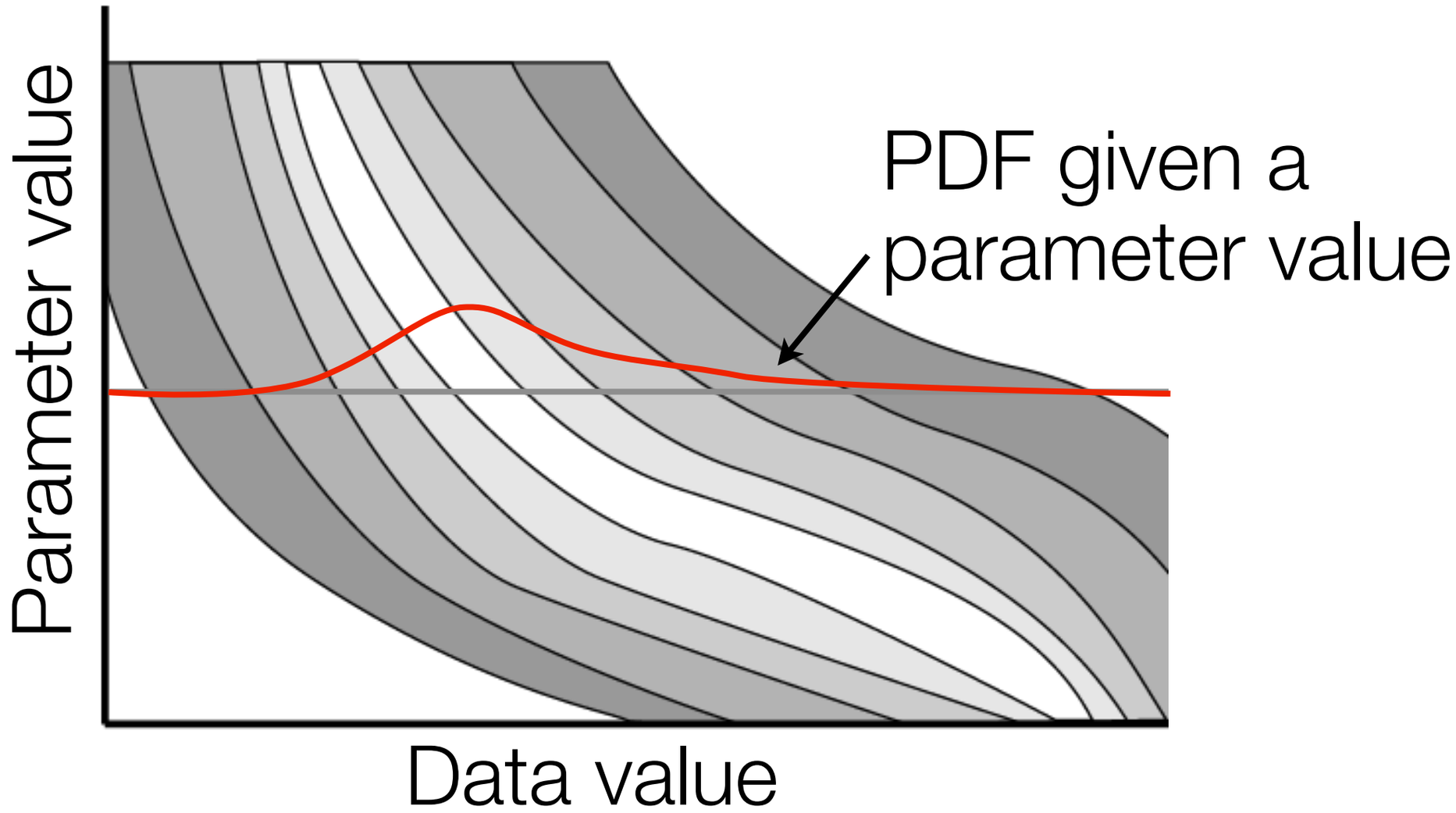
Likelihood Function



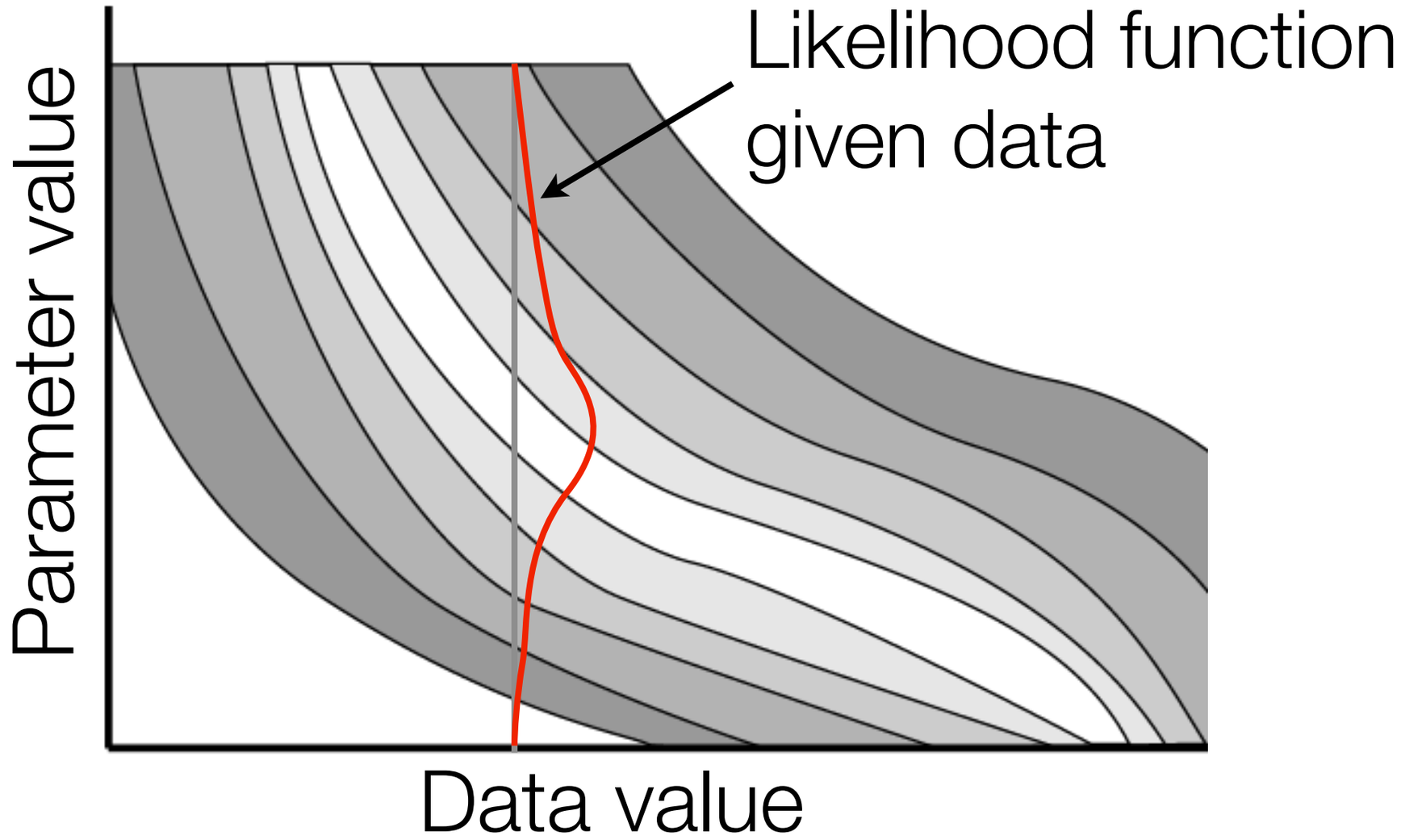
Likelihood Function



Likelihood Function



Likelihood Function



Maximum Likelihood

- **Consistency** - with sufficiently large number of observations n, it is possible to find the value of p with arbitrary precision (i.e. converges in probability to p)

- **Normality** - as the sample size increases, the distribution of the MLE tends to a Gaussian distribution with mean and covariance matrix equal to the inverse of the Fisher information matrix

- **Efficiency** - achieves CR bound as sample size$\longrightarrow\infty$ (no consistent estimator has lower asymptotic mean squared error than MLE)

Maximum likelihood recap

- In general, your likelihood is just the probability distribution of your data, assuming that you knew your parameter values
    - Then, we ‘re-think’ of that distribution as a function of the parameters with the data fixed—this is the likelihood

- Make the likelihood your cost function and find the parameter values that maximize it!
    - In other words, use optimization to figure out: what parameter values make your data very likely to be what the model would predict?
    - For some models, there are known formulas to find the optimum parameter values, but in many cases we have to use numerical (computational) optimization methods

How does this work with more complicated models?

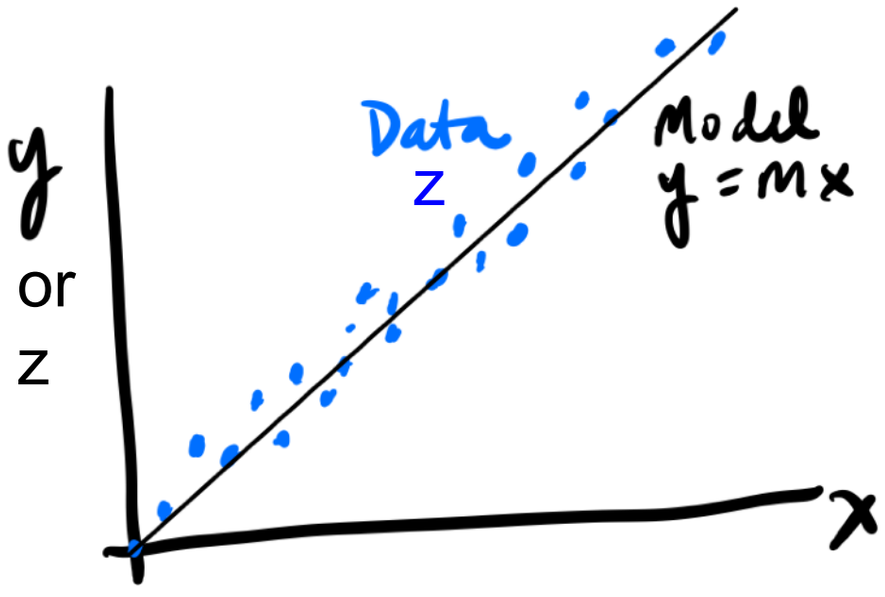
E.g. linear regression, etc?

Usually we view the model as describing one of the parameters or features/moments (e.g. the mean) of the distribution.

Revisit our least squares example—this is actually maximum likelihood!

Suppose we view the data (z) as coming from a normal distribution, but where the mean is given by a linear model like y = mx, and suppose we know the standard deviation

$$\begin{aligned} P(z_i|m) &= \mathcal{N}(\mu, \sigma) \\ &= \mathcal{N}(y_i, \sigma) \\ &= \mathcal{N}(mx_i, \sigma) \end{aligned}$$



Revisit code

Example: deterministic mean field model

- E.g. something like the Erdos-Renyi network SIR mean field model we derived before:

$$s_{t+1} = s_t - b\, s_t\, i_t$$

$$i_{t+1} = i_t + b\, s_t\, i_t - p_r\, i_t$$

$$r_{t+1} = (1 - s_t - i_t) + p_r\, i_t$$

- Or you can think of any other simple deterministic model if you'd prefer (e.g. some simple CA models would also work as the example here)

Example: difference equation model with Gaussian measurement error

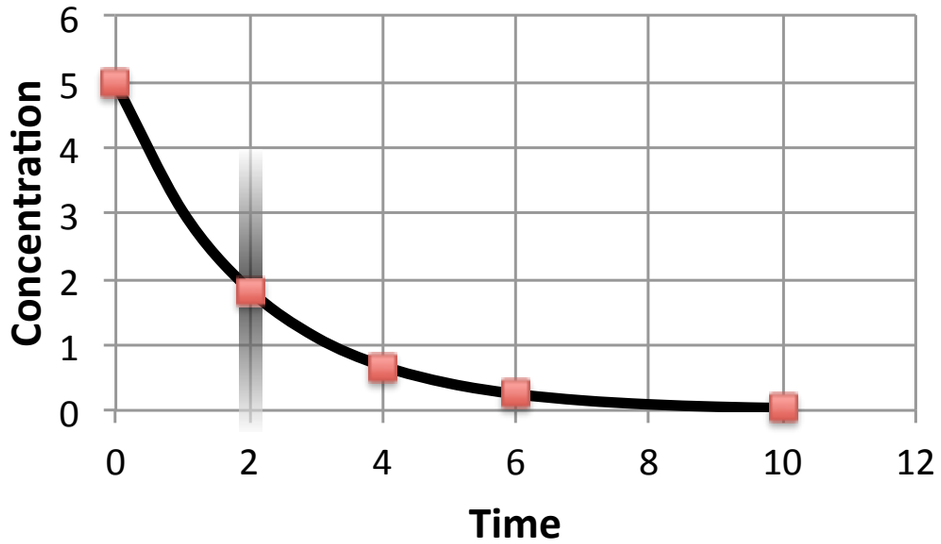
- Model:

$$x(t+1) = f(x,t,p)$$
$$y(t) = g(x,t,p)$$

- Discuss - measurement equation y
- Suppose data is taken at times $t_1, t_2, \ldots, t_n$
- Data at t_i = $z_i = y(t_i) + e_i$
- Suppose error is gaussian and unbiased, with known variance σ^2(can also be considered an unknown parameter)

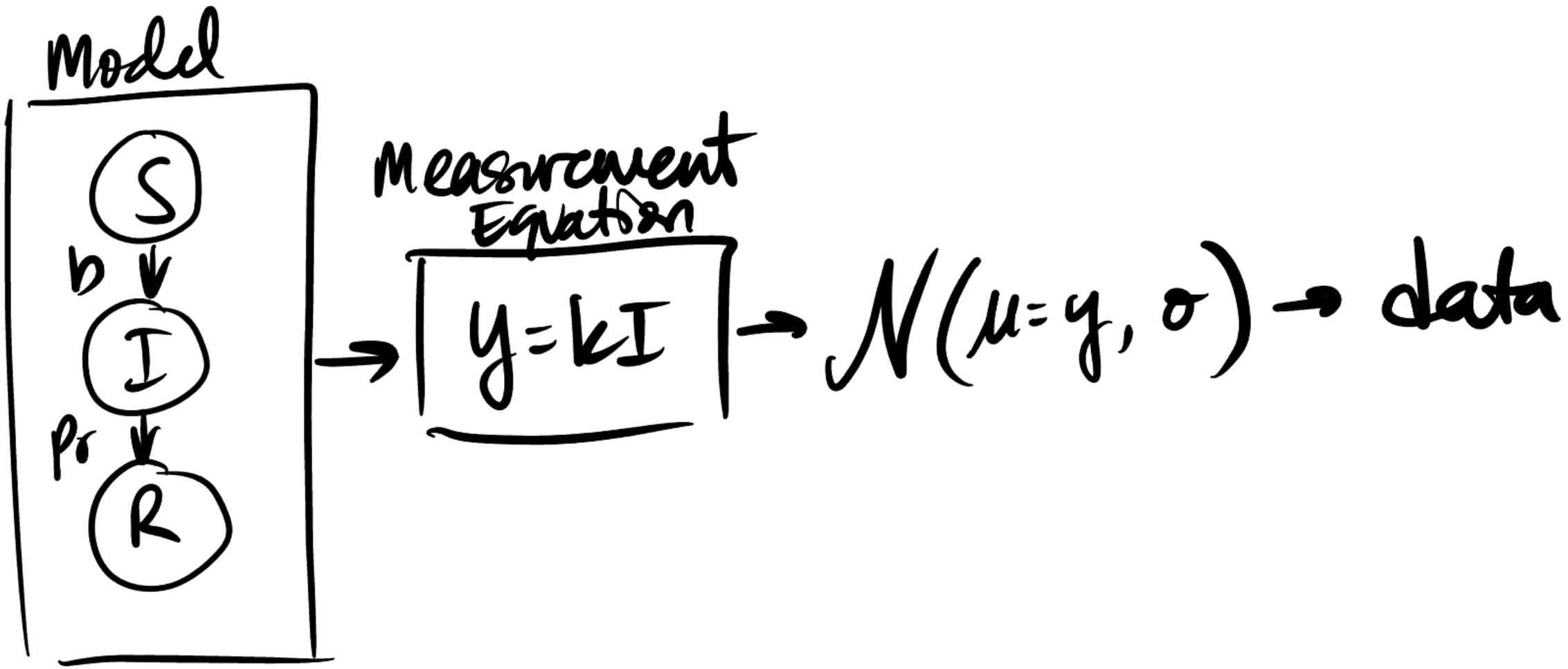
Example: difference equation model with Gaussian measurement error

- The measured data z_i at time i can be viewed as a sample from a Gaussian distribution with mean y(x, t_i,p) and variance σ^2



- Suppose all measurements are independent (is this realistic?)

Deterministic SIR Example



Example: difference equation model with Gaussian measurement error

- Then the likelihood function can be calculated as:

Gaussian PDF: $$f\left(z_i \mid \mu,\sigma^2\right) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(z_i-\mu\right)^2}{2\sigma^2}\right)$$

Example: difference equation model with Gaussian measurement error

- Then the likelihood function can be calculated as:

Gaussian PDF: $$f\left(z_i \mid \mu,\sigma^2\right) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(z_i-\mu\right)^2}{2\sigma^2}\right)$$

Formatted for model: $$f\left(z_i \mid y\left(x,t_i,p\right),\sigma^2\right) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(z_i-y\left(t_i,p\right)\right)^2}{2\sigma^2}\right)$$

Example: difference equation model with Gaussian measurement error

- Then the likelihood function can be calculated as:

Gaussian PDF: $$f\left(z_i \mid \mu,\sigma^2\right) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(z_i-\mu\right)^2}{2\sigma^2}\right)$$

Formatted for model: $$f\left(z_i \mid y\left(x,t_i,p\right),\sigma^2\right) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(z_i-y\left(t_i,p\right)\right)^2}{2\sigma^2}\right)$$

Likelihood function assuming independent observations:

$$L\left(y\left(t_i,p\right),\sigma^2 \mid z_1,\ldots,z_n\right) = f\left(z_1,\ldots,z_n \mid y\left(t_i,p\right),\sigma^2\right)$$

$$= \prod_{i=1}^{n} f\left(z_i \mid y\left(t_i,p\right),\sigma^2\right)$$

Example: difference equation model with Gaussian measurement error

$$L\left(y\left(t_i,p\right),\sigma^2 \mid z_1,\ldots,z_n\right) = f\left(z_1,\ldots,z_n \mid y\left(t_i,p\right),\sigma^2\right)$$

$$= \prod_{i=1}^{n} f\left(z_i \mid y\left(t_i,p\right),\sigma^2\right)$$

$$= \left(\frac{1}{2\pi\sigma^2}\right)^{n/2} \exp\left(-\frac{\sum_{i=1}^{n}\left(z_i - y\left(t_i,p\right)\right)^2}{2\sigma^2}\right)$$

Example: difference equation model with Gaussian measurement error

- It is often more convenient to minimize the Negative Log Likelihood (-LL) instead of maximizing the Likelihood
- Log is well behaved, minimization algorithms common

$$-LL = -\ln\left[\left(\frac{1}{2\pi\sigma^2}\right)^{n/2}\exp\left[-\frac{\sum_{i=1}^{n}\left(z_i - y\left(t_i, p\right)\right)^2}{2\sigma^2}\right]\right]$$

Example: difference equation model with Gaussian measurement error

$$-LL = -\ln\left[\left(\frac{1}{2\pi\sigma^2}\right)^{n/2}\exp\left[-\frac{\sum_{i=1}^{n}\left(z_i - y\left(t_i, p\right)\right)^2}{2\sigma^2}\right]\right]$$

$$-LL = -\left[-\frac{n}{2}\ln\left(2\pi\right) - n\ln\left(\sigma\right) - \frac{\sum_{i=1}^{n}\left(z_i - y\left(t_i, p\right)\right)^2}{2\sigma^2}\right]$$

Example: difference equation model with Gaussian measurement error

$$-LL = \frac{n}{2}\ln(2\pi) + n\ln(\sigma) + \frac{\sum_{i=1}^{n}\left(z_i - y(t_i, p)\right)^2}{2\sigma^2}$$

If σ is known, then first two terms are constants & will not be changed as p is varied—so we can minimize only the 3rd term and get the same answer

$$\min_p(-LL) = \min_p\left(\frac{\sum_{i=1}^{n}\left(z_i - y(t_i, p)\right)^2}{2\sigma^2}\right)$$

Example: difference equation model with Gaussian measurement error

- Similarly for denominator:

$$\min_p(-LL) = \min_p\left(\frac{\sum_{i=1}^{n}\left(z_i - y(t_i,p)\right)^2}{2\sigma^2}\right) = \min_p\left(\sum_{i=1}^{n}\left(z_i - y(t_i,p)\right)^2\right)$$

- This is just least squares!

- So, least squares is equivalent to the ML estimator when we assume a constant known variance

Maximum Likelihood

- Can calculate other ML estimators for different distributions
- Not always least squares-ish! (mostly not)
- Although surprisingly, least squares does fairly decently a lot of the time (more on this when we talk about approximate Bayesian computation)

Example - Poisson ML

- For count data (e.g. incidence data), the Poisson distribution is often more realistic than Gaussian

- Likelihood function?

Example - Poisson ML

- Model: $x(t+1) = f(x,t,p)$

$$y(t) = g(x,t,p)$$

- Data z_i is assumed to be Poisson with mean $y\left(t_i\right)$
- Assume all data points are independent
- Poisson PMF:

$$f\left(z_i \mid y\left(t_i\right)\right) = \frac{y\left(t_i\right)^{z_i}\, e^{-y\left(t_i\right)}}{z_i\,!}$$

Example - Poisson ML

- Likelihood function:

$$L\left(y\left(t,p\right)\mid z_1,\ldots,z_n\right) = f\left(z_1,\ldots,z_n\mid y\left(t,p\right)\right)$$

$$= \prod_{i=1}^{n} f\left(z_i\mid y\left(t,p\right)\right)$$

$$= \prod_{i=1}^{n} \frac{y\left(t_i\right)^{z_i}\, e^{-y\left(t_i\right)}}{z_i!}$$

Poisson ML

- Negative log likelihood:

$$-LL = -\ln\left(\prod_{i=1}^{n} \frac{y(t_i)^{z_i}\, e^{-y(t_i)}}{z_i!}\right)$$

$$= -\sum_{i=1}^{n} \ln\left(\frac{y(t_i)^{z_i}\, e^{-y(t_i)}}{z_i!}\right)$$

$$= -\sum_{i=1}^{n} z_i \ln\left(y(t_i)\right) + \sum_{i=1}^{n} y(t_i) + \sum_{i=1}^{n} \ln(z_i)$$

- Last term is constant

Example - Poisson ML

- Poisson ML Estimator:

$$\min_p(-LL) = \min_p\left(-\sum_{i=1}^{n} z_i \ln\left(y(t_i)\right) + \sum_{i=1}^{n} y(t_i)\right)$$

- Other common distributions - negative binomial (overdispersion), zero-inflated poisson or negative binomial, etc.

Maximum likelihood for deterministic models

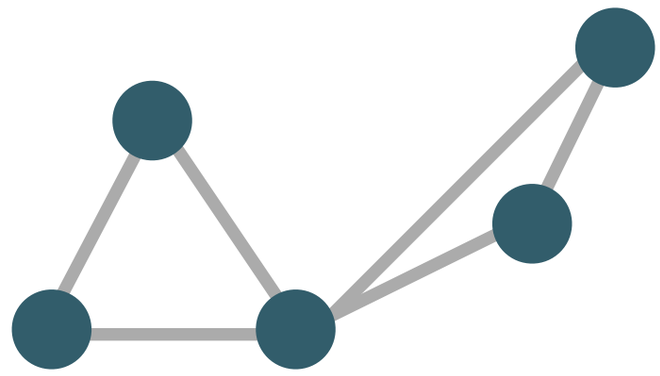
- Basic approach for deterministic models - suppose only measurement error (otherwise distribution is determined by the model stochasticity and measurement error)
- Data is given by distribution where model output is the mean
- Suppose each time point of data is independent
- Use PDF/PMF to calculate the likelihood
- Take the negative log likelihood, minimize this over the parameter space

Maximum Likelihood for ABMs & other kinds of models

- Can be quite different!
- May require more computation to evaluate (e.g. stochastic models)
- May also be structured quite differently! (e.g. network or individual-based models)

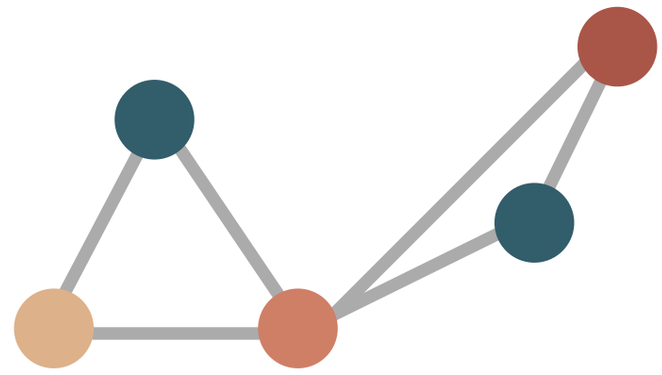
Tiny Network Example

- Data: infection pattern on the network
- Model: suppose constant probability *p* of infecting along an edge from someone who got sick before you
- What's the likelihood?



Tiny Network Example

- Data: infection pattern on the network
- Model: suppose constant probability *p* of infecting along an edge, assuming we start with first case
- What's the likelihood?
- Let's think about how we would calculate it for a specific data set



- L(p,data) = P(susc nodes did not get sick)
  x P(infected nodes did get sick)

(note assumes independence, not realistic!)

Sampling-based approaches to parameter estimation

- In our maximum likelihood example, we were able to write down our likelihood explicitly, in terms of equations (e.g. using a normal distribution and the model equations)

- However, for more complex models, or for Bayesian estimation, it's often difficult or impossible to write down an equation for the posterior/likelihood/etc.

Sampling-based approaches to parameter estimation

- Instead—we can use sampling based approaches to sample from the posterior/likelihood—this is often more tractable for ABMs and other complex models

- More on this next time!

Some points when you're doing parameter estimation

- People will often just run the optimization and then accept the parameter estimates that come out of it—these can be totally meaningless (e.g. if the fit is bad, or you have unidentifiability)

- Be sure visualize the fit of your model! Plot the model and the data on the same plot and see how closely they match

- Visualization can be difficult with high dimensional data sets! You may need to do some thinking about how to visualize the model fit to data

Some points when you're doing parameter estimation

Be careful about local minima or canyons (unidentifiability)!

- Optimization algorithms will not always know or warn you that this is a problem—you can be very mislead by parameter estimates if this is the case

Some points when you're doing parameter estimation

Think carefully about how you choose your model!

- Assumptions, overfitting, and model selection
- Try multiple models (and think carefully about how you will choose between them!)