Lecture 9: Computational Cognitive Modeling

Model Fitting, Estimation, and Comparison

course website:
https://brendenlake.github.io/CCM-site/

What makes a good model?

Qualitative Criteria

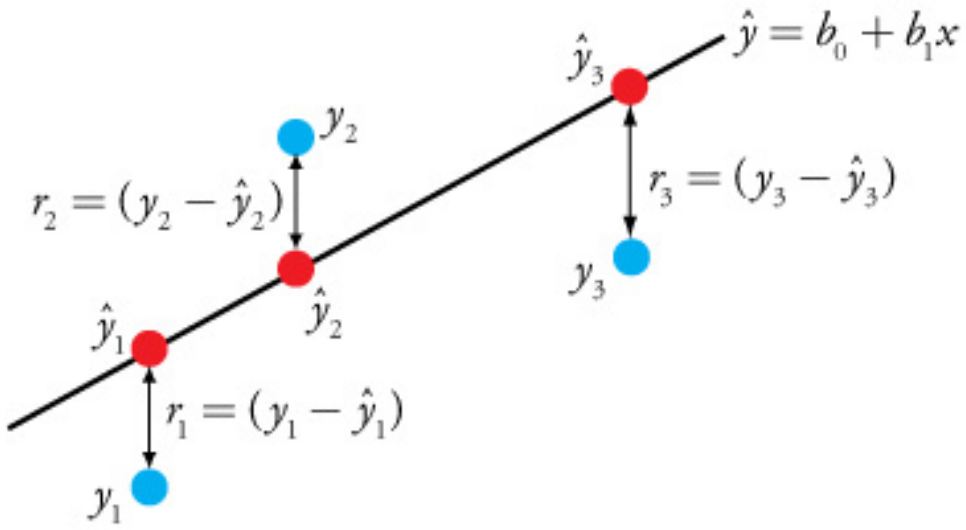
- Explanatory adequacy
  - The assumptions of the model are plausible and consistent with other findings (assumption are not ad-hoc)
- The model does more than just re-describe the data
  - “curve fitting” versus theoretical principals
  - e.g., “power law of practice” versus “instance theory of automaticity”
- Interpretability
  - The model makes sense
  - The model components and parameters link to psychological or neural processes and constructs

Qualitative Criteria

- Faithful
  - model depends on “core” theoretical features, not implementation details
    - e.g., neural network shouldn’t depend on relative number of “hidden units” unless a strong commitment is being made
- Parsimony?
  - Occam’s razor?

Quantitative Criteria

- Goodness-of-fit
  - SSE, RMSE, etc...
  - A good fit merely qualifies the model as one of the candidate models for further consideration... necessary but not sufficient.



Quantitative Criteria

Table I. Results of a model recovery simulation in which a GOF measure (RMSE) was used to discriminate models when the source of the error was varied.

Condition (sources of variation)	Model the data were generated from			Model fitted	
	M_A $a = 0.4$	M_A $a = 0.6$	M_B	M_A	M_B
(1) Sampling error	100	–	–	0.040 (0%)	0.029 (100%)
(2) Sampling error + individual differences	50	50	–	0.041 (0%)	0.029 (100%)
(3) Different models	–	50	50	0.075 (0%)	0.029 (100%)
(4) Sampling error	–	–	100	0.079 (0%)	0.029 (100%)

The severity of the problem is shown in Table I, which contains the results of a model recovery simulation using RMSE. Four datasets were generated from a combination of the two models (M_A and M_B), defined as follows: $M_A: y = (1+t)^{-a}$, $M_B: y = (b+ct)^{-a}$ where $a, b, c > 0$. Datasets

Quantitative Criteria

- Goodness-of-fit
  - SSE, log(L), RMSE, etc...
  - A good fit merely qualifies the model as one of the candidate models for further consideration... necessary but not sufficient.
- Parsimony?
  - The “simplest” model that does not fit significantly worse than possibly more complex models
  - via Heirarchical model testing, G^2 statistics, etc...
- Generalizability
  - model can account for new unseen data

Quantitative Criteria

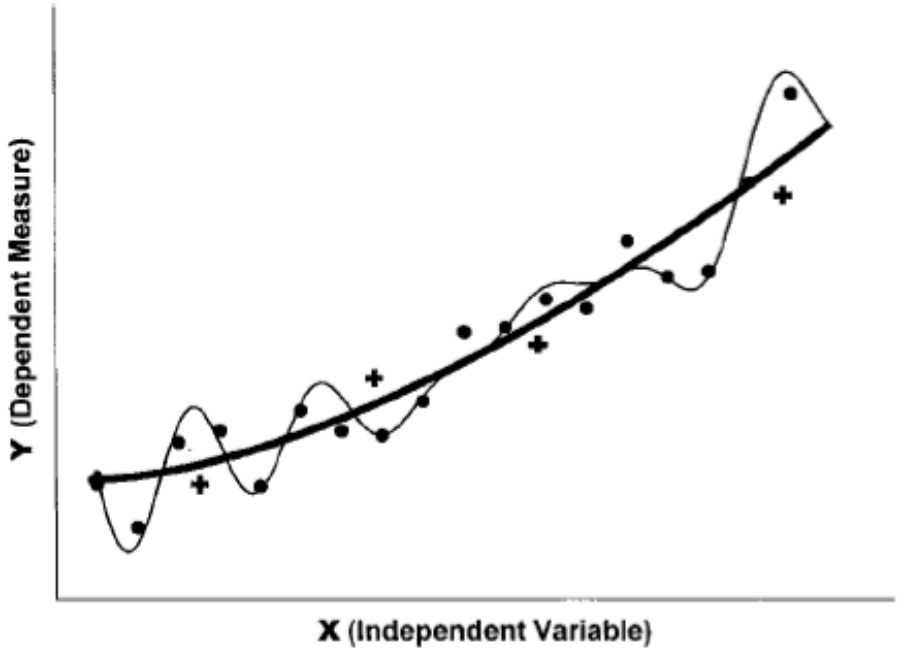


Figure 1. Illustration of the trade-off between goodness of fit and generalizability. An observed data set (dots) was fitted to a simple model (thick line) and a complex model (thin line). New observations are shown by the plus symbol.

Quantitative Criteria

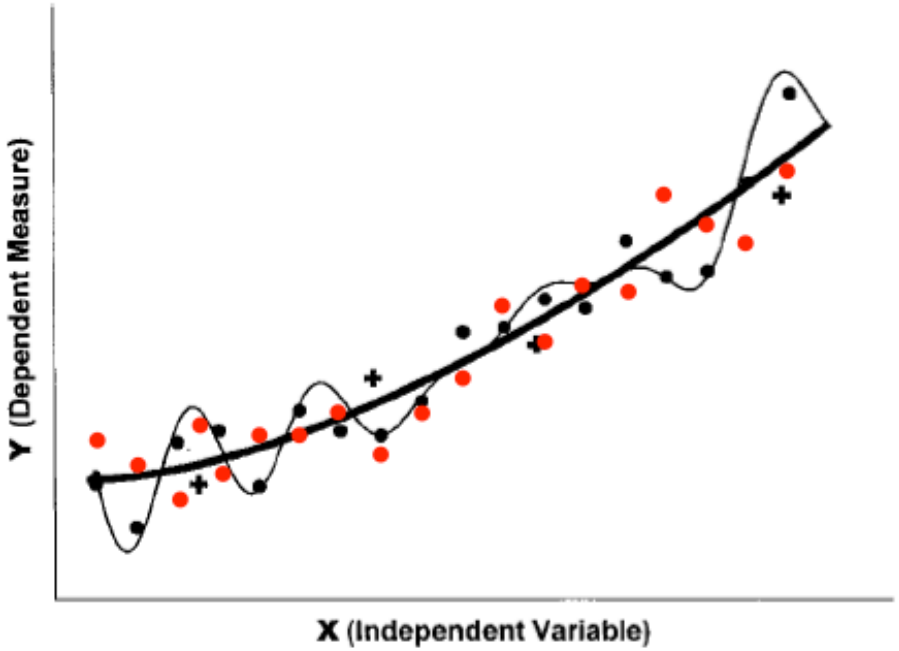


Figure 1. Illustration of the trade-off between goodness of fit and generalizability. An observed data set (dots) was fitted to a simple model (thick line) and a complex model (thin line). New observations are shown by the plus symbol.

Quantitative Criteria

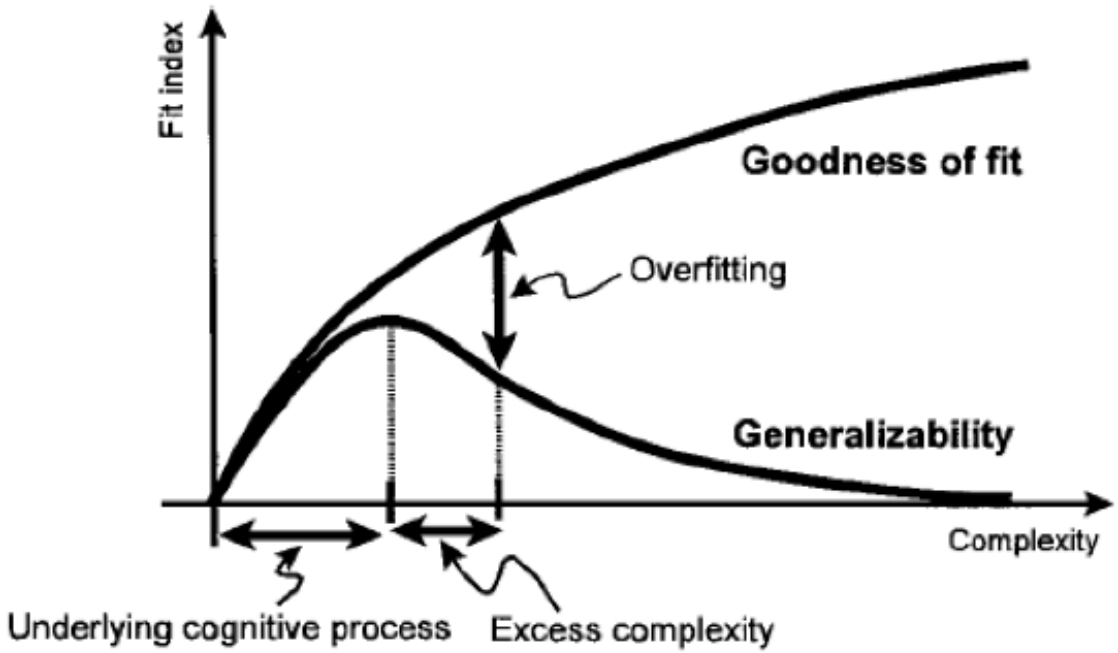


Figure 2. Illustration of the relationship between goodness of fit and generalizability as a function of model complexity (Myung & Pitt, 2001). From *Stevens' Handbook of Experimental Psychology* (p. 449, Figure 11. 4), by J. Wixted (Editor), 2001, New York: Wiley. Copyright 2001 by Wiley. Adapted with permission.

Quantitative Criteria

- Complexity
  - Inherent flexibility in model
  - AIC, BIC - # of free parameters (basically goodness of fit+ penalty)
  - insensitive to the “functional form” of the model

Quantitative Criteria

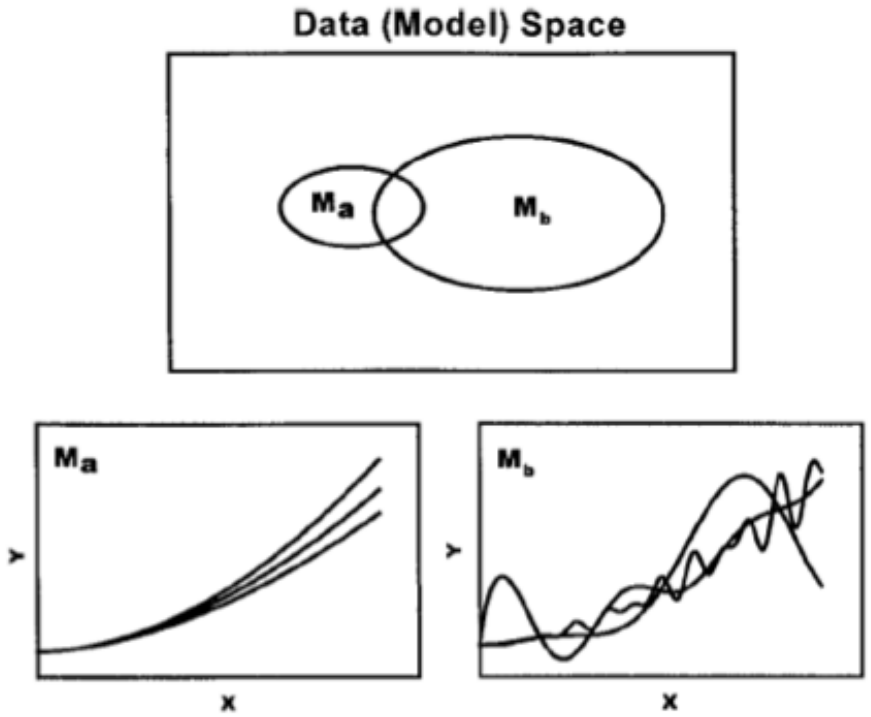
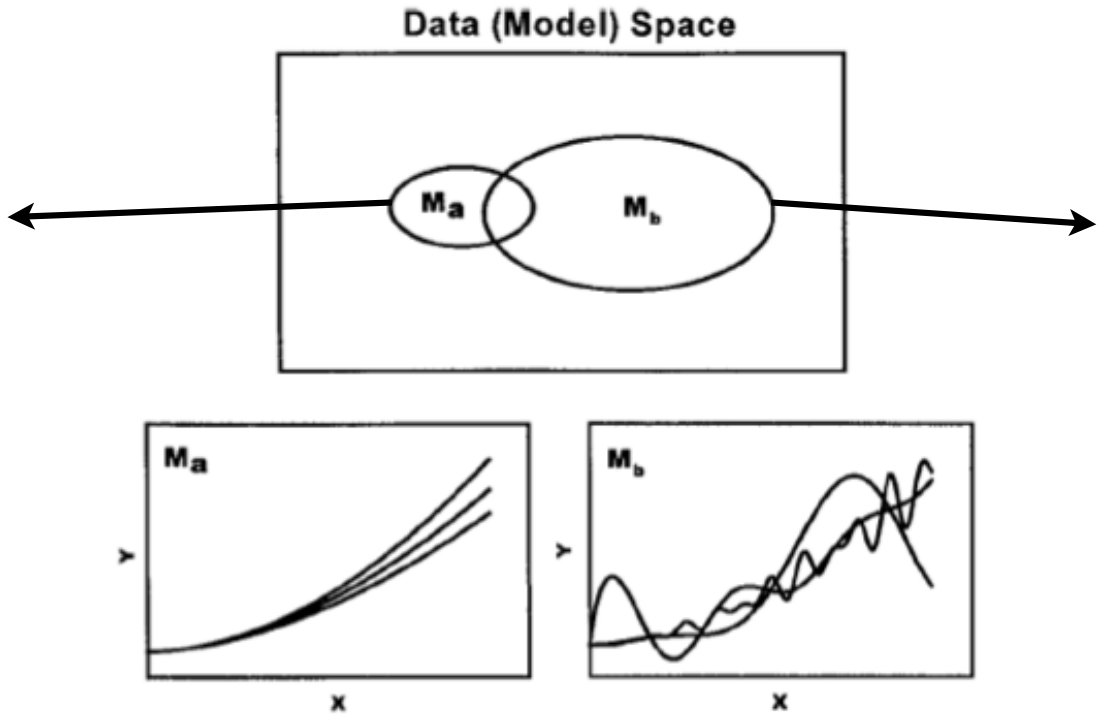


Figure 3. The top panel depicts regions in data space occupied by two models, M_a (simple model) and M_b (complex model), with the range of data patterns that can be generated by each model in the lower panels.

Quantitative Criteria

simple model (easy to falsify)



complex model (harder to falsify)

Figure 3. The top panel depicts regions in data space occupied by two models, M_a (simple model) and M_b (complex model), with the range of data patterns that can be generated by each model in the lower panels.

Quantitative Criteria

Table II. Two GOF Measures, four generalizability measures, and the dimensions of complexity to which each is sensitive

Selection method	Criterion equation	Dimensions of complexity considered
Root Mean Squared Error	$RMSE = (SSE/N)^{1/2}$	None
Percent Variance Accounted For	$PVAF=100(1-SSE/SST)$	None
Akaike Information Criterion	$AIC = -2\ ln(f(y\|\theta_0)) + 2k$	Number of parameters
Bayesian Information Criterion	$BIC = -2\ ln(f(y\|\theta_0)) + k \cdot ln(n)$	Number of parameters, sample size
Bayesian Model Selection	$BMS=-ln \int f(y\|\theta)\pi(\theta)d\theta$	Number of parameters, sample size, functional form
Minimum Description Length	$MDL=-ln\,(f(y\|\theta_0)) + (k/2)ln(n/2\pi)+ln \int \sqrt{\det(I(\theta))}d\theta$	Number of parameters, sample size, functional form

In the equations above, y denotes observed data, θ is the model's parameter, θ_0 is the parameter value that maximizes the likelihood function $f(y|\theta)$, k is the number of parameters, n is the sample size, N is the number of data points fitted, SSE is the minimized sum of the squared errors between observations and predictions, SST is the sum of the squares total, $\pi(\theta)$ is the parameter prior density, $I(\theta)$ is the Fisher information matrix in mathematical statistics [a], *det* denotes the determinant of a matrix, and *ln* denotes the natural logarithm of base e.

Quantitative Criteria

Cognitive Science 34 (2010) 10–50

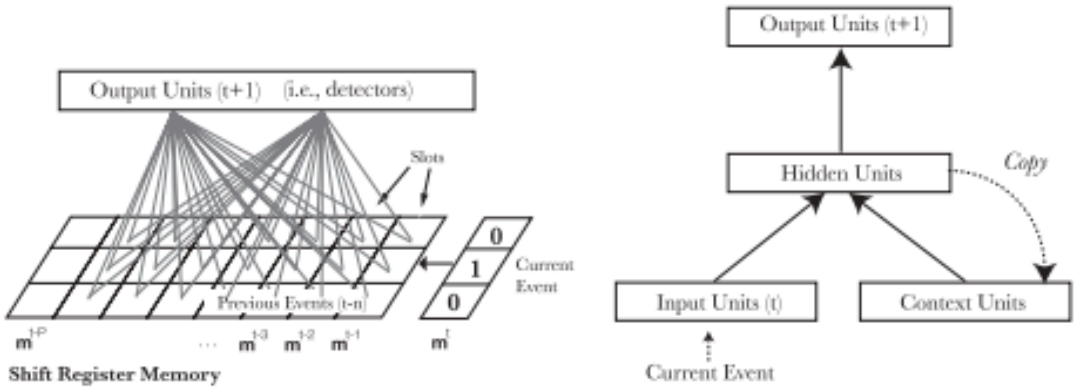
ISSN: 0364-0213 print / 1551-6709 online
DOI: 10.1111/j.1551-6709.2009.01076.x

Direct Associations or Internal Transformations? Exploring the Mechanisms Underlying Sequential Learning Behavior

Todd M. Gureckis,[a] Bradley C. Love[b]

[a]*Department of Psychology, New York University*
[b]*Department of Psychology, The University of Texas at Austin*

Quantitative Criteria



Linear Associative Shift-Register (LASR) **Elman Network (SRN)**

Fig. 1. The schematic architecture of the LASR (left) and SRN (right) networks. In LASR, memory takes the form of a shift-register. New events enter the register on the right and all previous register contents are shifted left by one position. A single layer of detector units learns to predict the next sequence element given the current contents of the register. Each detector is connected to all outcomes at all memory slots in the register. The model is composed of *N* detectors corresponding to the *N* event outcomes to be predicted (the weights for only two detectors is shown). In contrast, in the SRN, new inputs are presented on a bank of input units and combine with input from the context units to activate the hidden layer (here, solid arrows reflect fully connected layers, and dashed arrows are one-to-one layers). On each trial, the last activation values of the hidden units are copied back to the context units, giving the model a recurrent memory for recent processing. In both models, learning is accomplished via incremental error-driven adaptation of learning weights.

Quantitative Criteria

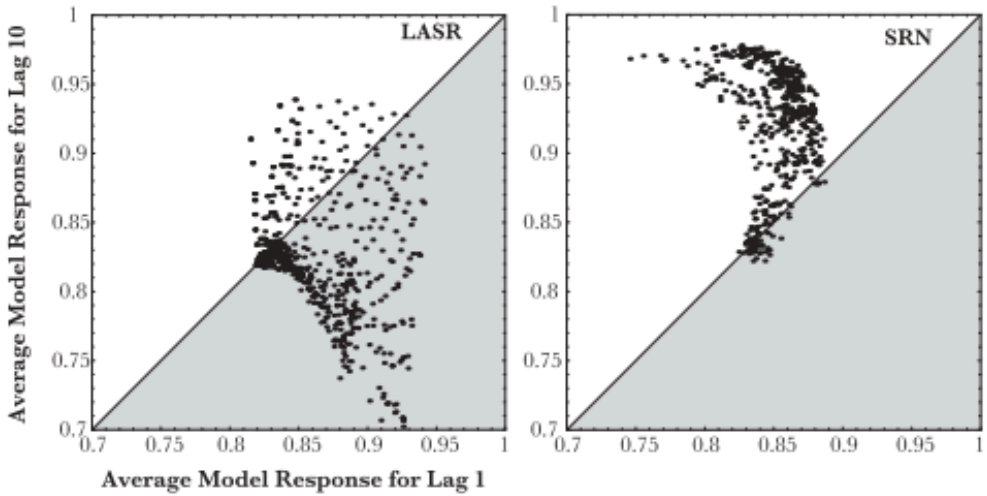


Fig. 9. Explorations of the parameter space for LASR and the SRN in Experiment 1. Each model's average response for lag-1 is plotted against the average response for lag-10. The division between the grey and white regions represents the line $y = x$. Each point in the plot represents the performance of the respective model with a particular setting of the parameters. If the point appears below the $y = x$ line in the grey area, it means the model predicts faster responding to lag-10 events than to lag-1 (the correct qualitative pattern). Note, however, that accounting for the full pattern of human results requires a monotonically decreasing function of predicted RT across all 10 event lags, while this figure only illustrates the two end points (lag-1 and lag-10). Thus, the few instances where the SRN appears to predict the correct pattern are not in general support for the model (see main text).

Quantitative Criteria

people show learning only in this condition (and so does LASR)

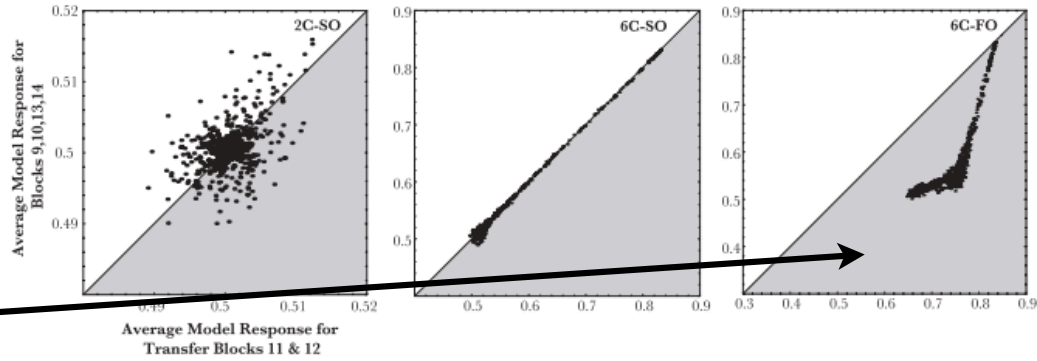


Fig. 11. Explorations of the parameter space of the LASR in the three conditions (2C-SO, 6C-SO, 6C-FO) of Experiment 2. The key behavioral pattern from Experiment 2 was that subjects only responded differently during transfer block in the 6C-FO condition. The model's average response for the transfer blocks 11 and 12 are plotted against the average response for the surrounding learning block (9, 10, 13, and 14). Also plotted is the line $y = x$. Each point in the plot represents the performance of the LASR with a particular setting of the parameters. If the point appears below the $y = x$ line, it means the model predicts slower responding during the transfer blocks (and thus evidence of learning). LASR predicts a systematic learning effect only in the 6C-FO condition (i.e., first-order, linear learning), like human subjects.

SRN shows learning in all conditions!

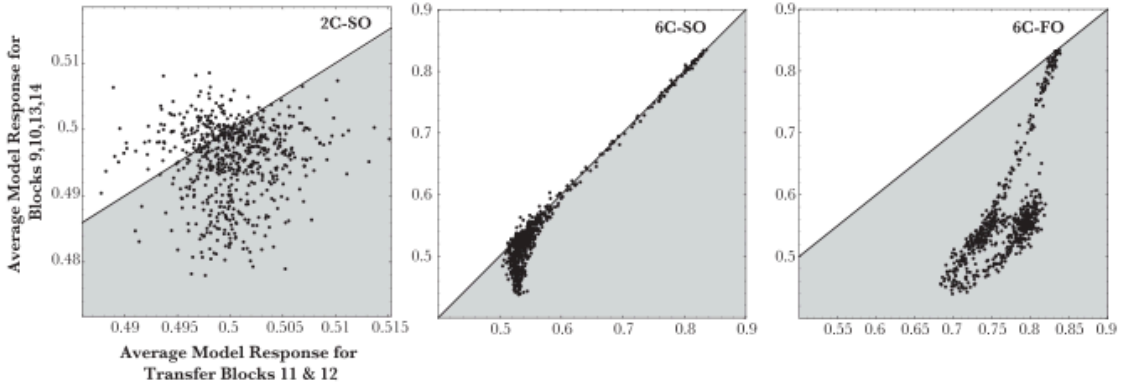


Fig. 12. Explorations of the parameter space of the SRN in the three conditions (2C-SO, 6C-SO, 6C-FO) of Experiment 2. The model's average response for the transfer blocks (11 and 12) are plotted against the average response for the surrounding learning block (9, 10, 13, and 14). Also plotted is the line $y = x$. Each point in the plot represents the performance of the SRN with a particular setting of the parameters. If the point appears below the $y = x$ line, it means the model predicts slower responding during the transfer blocks (and thus a learning effect). The SRN, unlike human subjects, shows a learning effect in all three conditions.

Parameter fitting techniques

How do we find the values of model parameters that maximize the fit of a model to observed data?

we need some

“measure of fit”

Examples:

1. Pearson Correlation
2. Sum-squared error
3. Root mean squared error
4. % Variance accounted for
5. Likelihood

we need some

“measure of fit”

Examples:

1. Pearson Correlation
2. Sum-squared error
3. Root mean squared error
4. % Variance accounted for
5. Likelihood

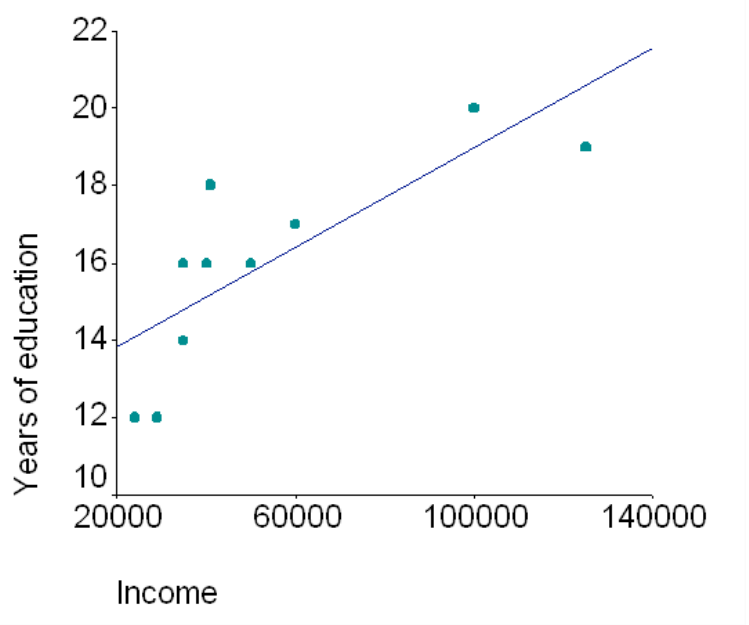
goodness of fit measures

ironically LESS GOOD!

better!

pearson correlation

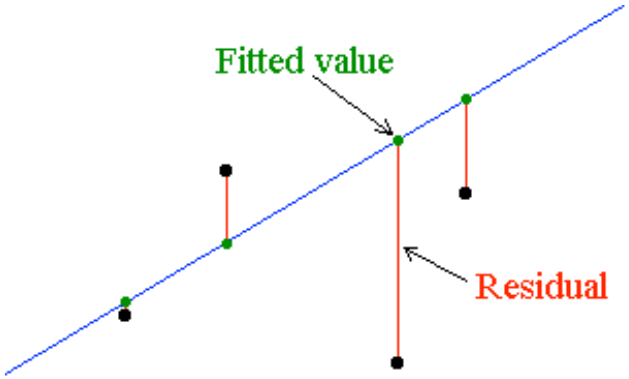
$$r_{obs,prd} = \frac{\sum (obs - \mu_{obs})(prd - \mu_{prd})}{\sqrt{\sum (obs - \mu_{obs})^2 \sum (prd - \mu_{prd})^2}}$$



everyone is likely very familiar with this!

sum of squared error (SSE)

$$SSE_{obs,prd} = \sum (obs - prd)^2$$



nothing new here.

root mean squared error (RMSE)

$$RMSE_{obs,prd} = \sqrt{\frac{\sum(obs - prd)^2}{N}}$$

used this last time.

% variance

$$\%Var = \frac{SSE_{null} - SSE_{model}}{SSE_{null}}$$

$$SSE_{null} = \sum_i (obs_i - \mu_{obs})^2$$

$$SSE_{model} = \sum_i (obs_i - prd_i)^2$$

likelihood

$$p(d|model)$$

more on this in a bit

ok, so how to find good parameters?

1. Calculus
2. Grid search
3. Hill climbing
4. Nelder-meade simplex (used by fmin in python and matlab)
5. Simulated Annealing
6. Genetic algorithms

calculus

d_ij	obs s_ij	prd s_ij
0	1	
1	0.368	
2	0.135	
3	0.05	
4	0.018	
5	0.007	

$$s_{ij} = \alpha + \beta d_{ij}$$

find parameters for regression that minimize SSE between obs s_ij and predicted sij

calculus

$$SSE = \sum_k (obs_k - prd_k)^2$$

$$SSE = \sum_k (obs_k - (\alpha + \beta d_k)^2$$

calculus

$$SSE = \sum_k (obs_k - prd_k)^2$$

$$SSE = \sum_k (obs_k - (\alpha + \beta d_k)^2$$

$$\frac{\partial SSE}{\partial \alpha} = \sum_k 2(obs_k - \alpha - \beta d_k)(-1)$$

$$\frac{\partial SSE}{\partial \beta} = \sum_k 2(obs_k - \alpha - \beta d_k)(-d_k)$$

$$\frac{\partial SSE}{\partial \alpha} = -2\sum_k (obs_k - \alpha - \beta d_k)$$

$$\frac{\partial SSE}{\partial \beta} = -2\sum_k d_k(obs_k - \alpha - \beta d_k)$$

calculus

$$SSE = \sum_k (obs_k - prd_k)^2$$

$$SSE = \sum_k (obs_k - (\alpha + \beta d_k)^2$$

$$\frac{\partial SSE}{\partial \alpha} = \sum_k 2(obs_k - \alpha - \beta d_k)(-1)$$

$$\frac{\partial SSE}{\partial \beta} = \sum_k 2(obs_k - \alpha - \beta d_k)(-d_k)$$

$$\frac{\partial SSE}{\partial \alpha} = -2\sum_k (obs_k - \alpha - \beta d_k)$$

$$\frac{\partial SSE}{\partial \beta} = -2\sum_k d_k(obs_k - \alpha - \beta d_k)$$

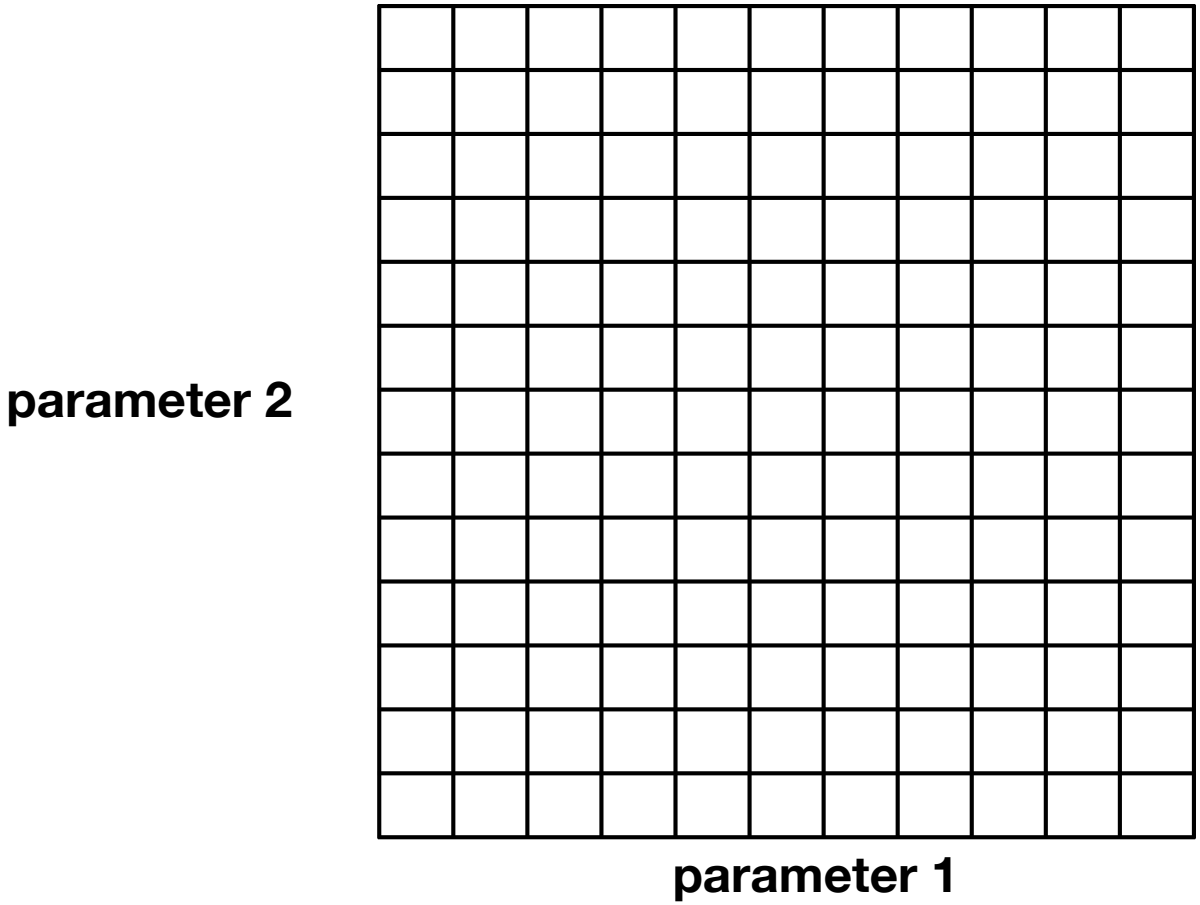
$$\frac{\partial SSE}{\partial \alpha} = 0$$

$$\frac{\partial SSE}{\partial \beta} = 0$$

ok, who want to write this down for the deep q-learning network?

in general, when you aren't smart enough, simulate!

grid search



calculated SSE at each combination of parameters and keep the best.

grid search

```
from scipy.optimize import brute

brute(evalmodel, [[min, max], [min, max]], args, Ns=20, full_output=1)
```

grid search

in general, this sucks.

evaluation time for one set of parameters

x

# of evaluations

grid search

in general, this sucks.

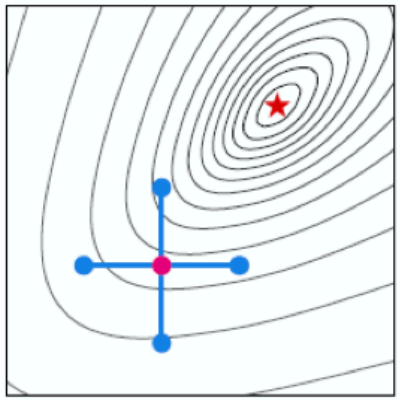
100 seconds per evaluation

x

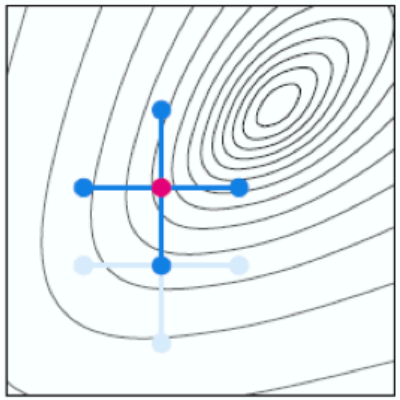
10^12 of evaluations

10^14 seconds = 3 million years!

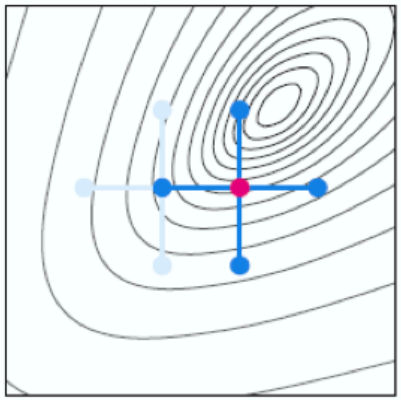
hill climbing



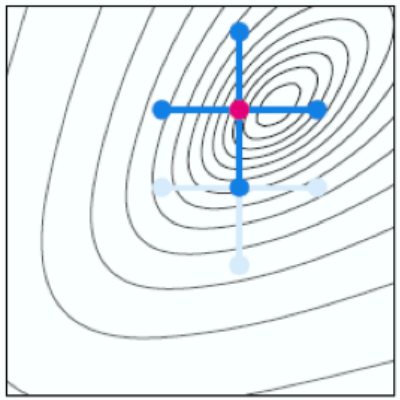
(a) Initial pattern



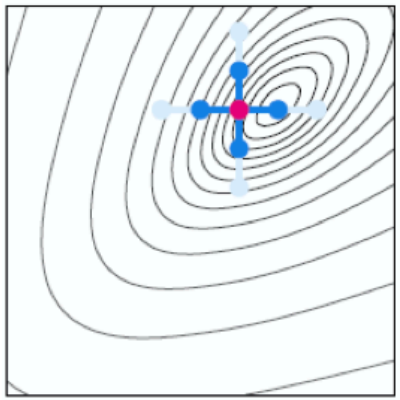
(b) Move North



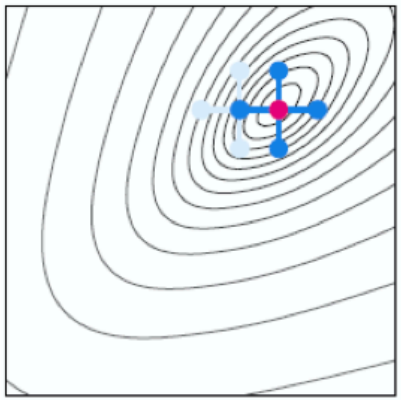
(c) Move West



(d) Move North



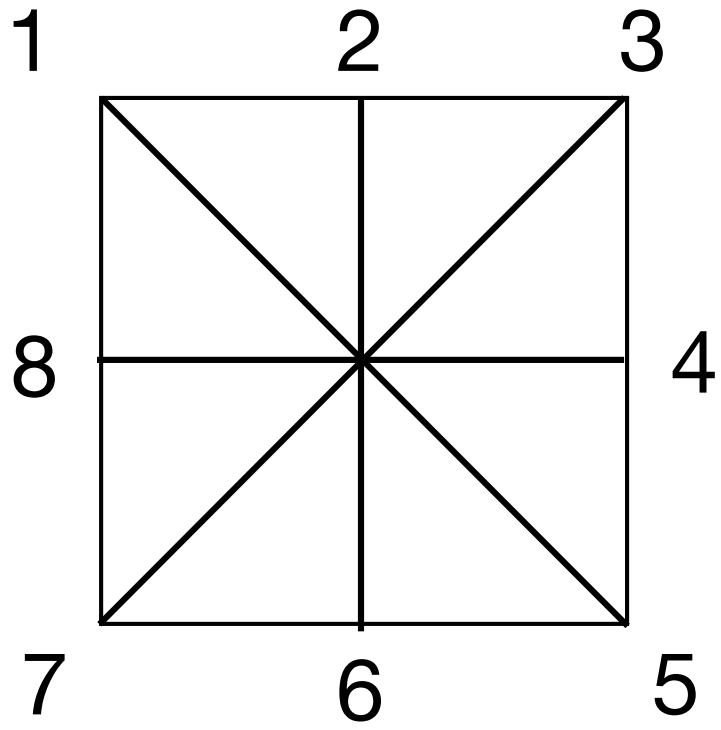
(e) Contract



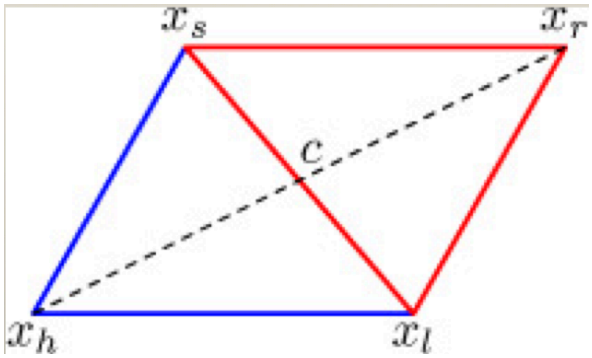
(f) Move West

hill climbing

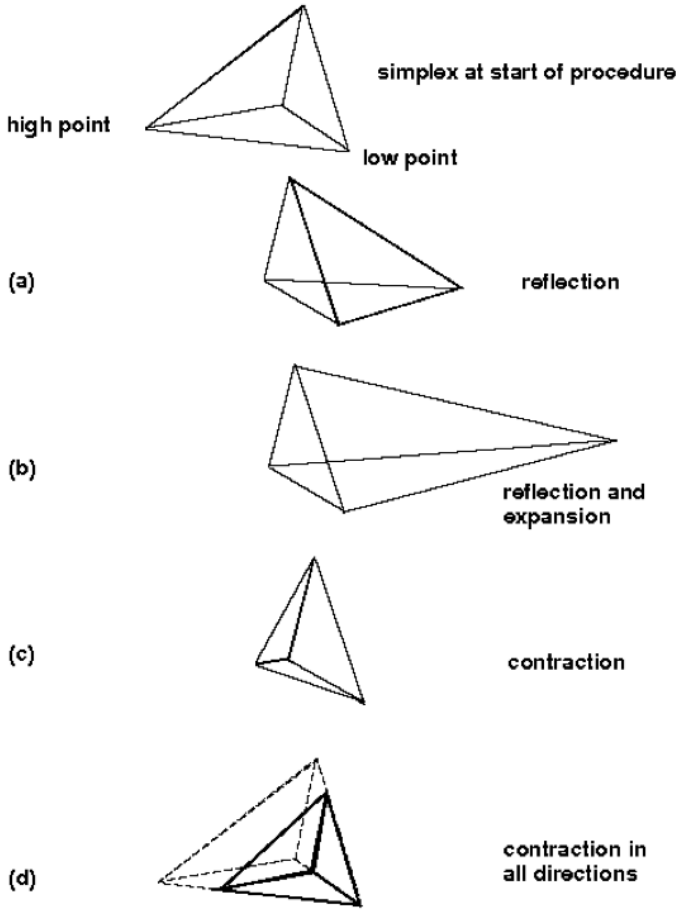
how many points are evaluated each step?



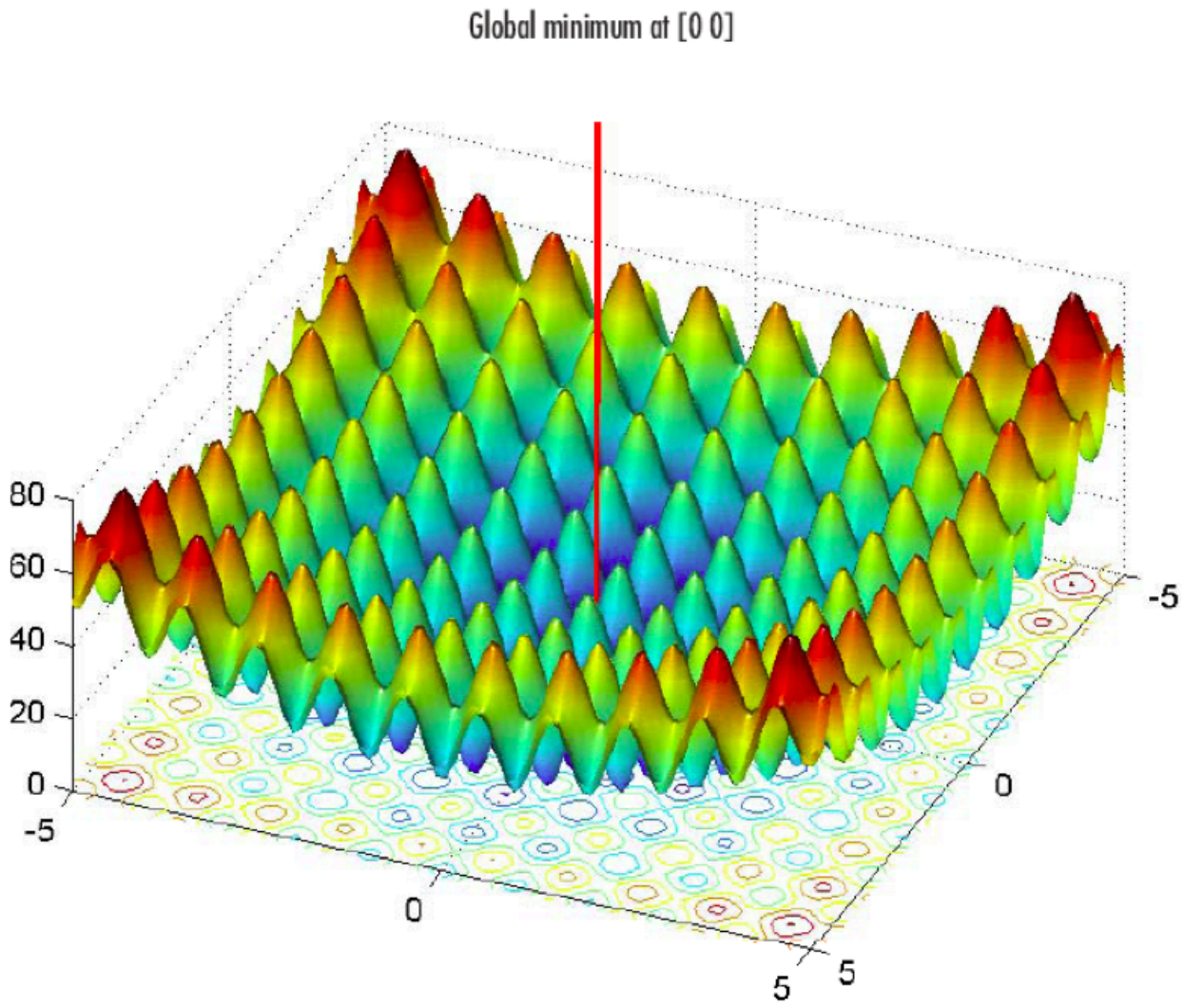
nelder mead simplex



the simplex approach minimizes the number of points you have to evaluate



start multiple places!



does your search code work?

how do you know?

does your search code work?

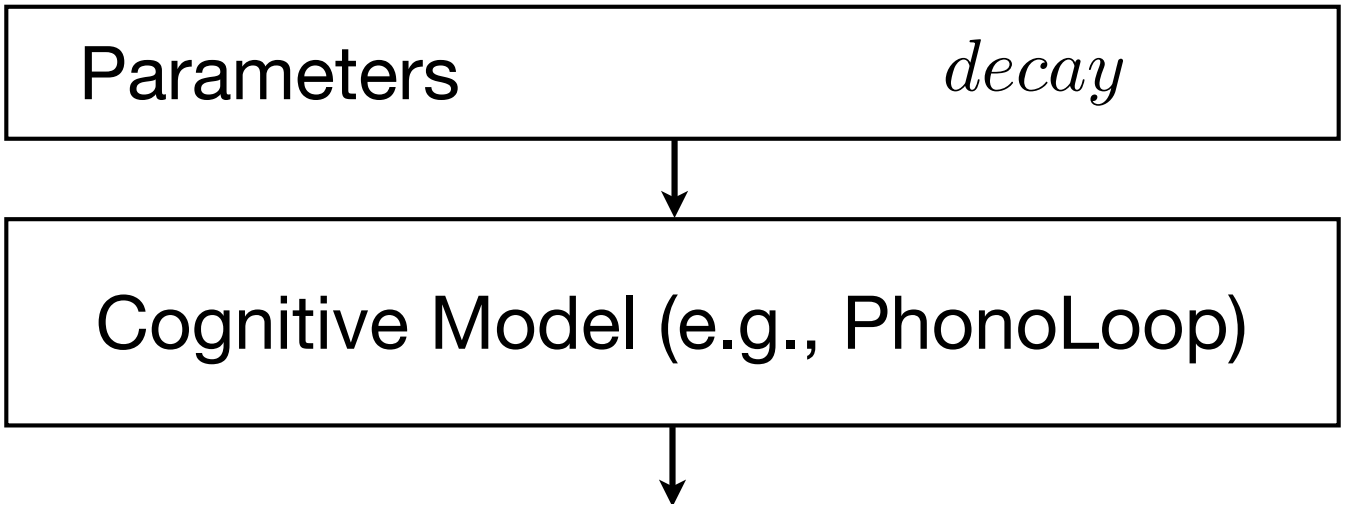
One idea:

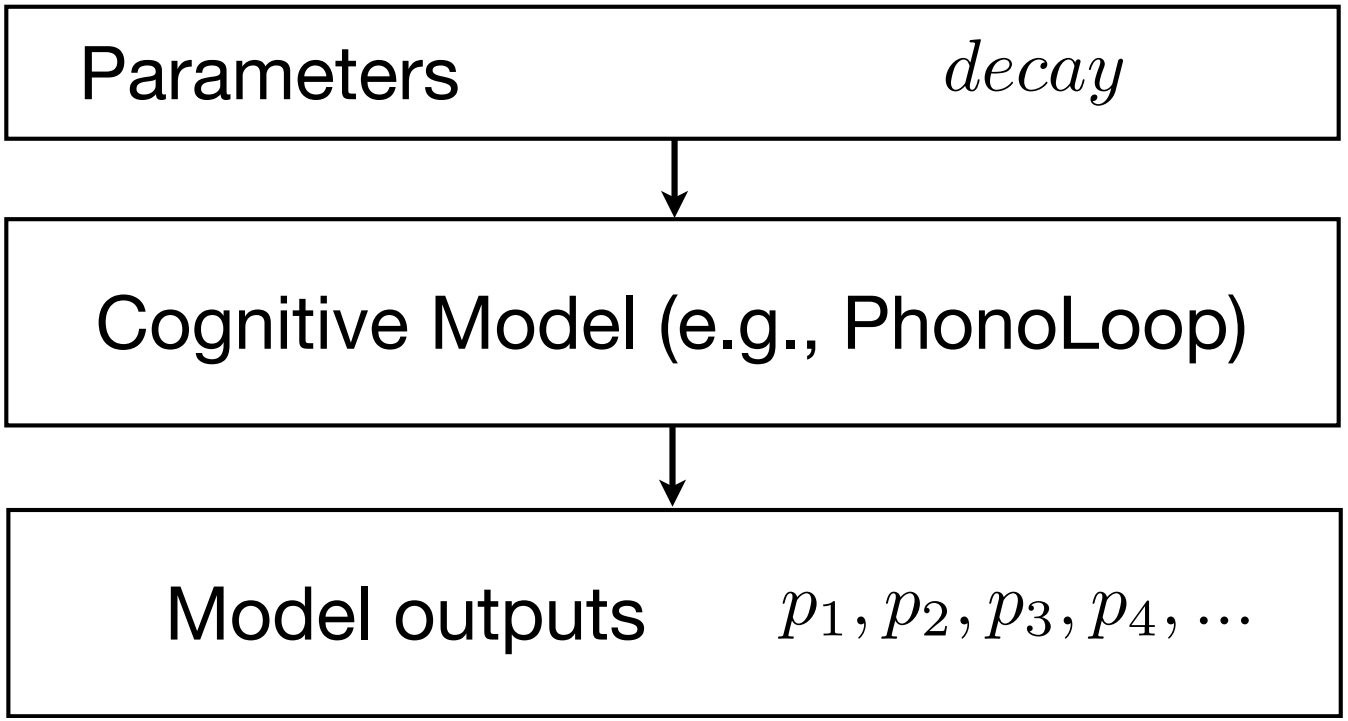
Run your model with known parameters, to GENERATE pretend human data. Then, re-fit the model using your fit measure and parameter search algorithm to see if you can get the same parameters you used to generate the data.

If this doesn't work the model parameters may be non-identifiable (multiple parameters give the same fit).

Maximum likelihood estimation

- Myung (2003) - “MLE is not as widely recognized among modelers in psychology, but it is a standard approach to parameter estimation and inference in statistics.”
- Sufficiency - complex information about parameter of interest contained in MLE estimator
- Consistency - true parameter value that generated the data can be recovered asymptotically
- Parameter invariance - same MLE solution obtained independent of way parameterized
- LSE is a hack, but is correct for linear regression. But shouldn’t be used all over the place.


Parameters
$decay$
Cognitive Model (e.g., PhonoLoop)


Parameters
$decay$
Cognitive Model (e.g., PhonoLoop)
Model outputs
$p_1, p_2, p_3, p_4, \cdots$

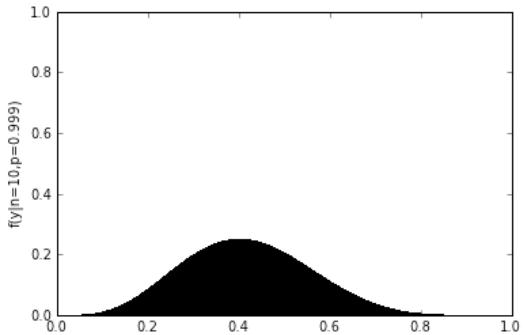
Parameters $decay$

Cognitive Model (e.g., PhonoLoop)

Model outputs $p_1, p_2, p_3, p_4, \cdots$

Likelihood model

$$P(X = k) = \binom{n}{k} p^k (1.0 - p)^{(n-k)}$$



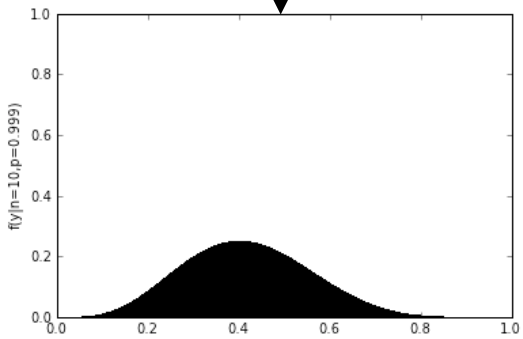
Parameters $decay$

Cognitive Model (e.g., PhonoLoop)

Model outputs $p_1, p_2, p_3, p_4, \cdots$

Likelihood model

$$P(X = k) = \binom{n}{k} p^k (1.0 - p)^{(n-k)}$$



$$L(\theta|d) = \prod^{k} L(\theta|d_k)$$

Data $d = \frac{k_1}{n_1}, \frac{k_2}{n_2}, \frac{k_3}{n_3}, \frac{k_4}{n_4}, \ldots$

Search algorithm `fmin`

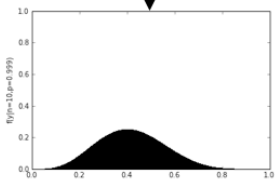
Parameters $decay$

Cognitive Model (e.g., PhonoLoop)

Model outputs $p_1, p_2, p_3, p_4, \cdots$

Likelihood model

$$P(X = k) = \binom{n}{k} p^k (1.0 - p)^{(n-k)}$$

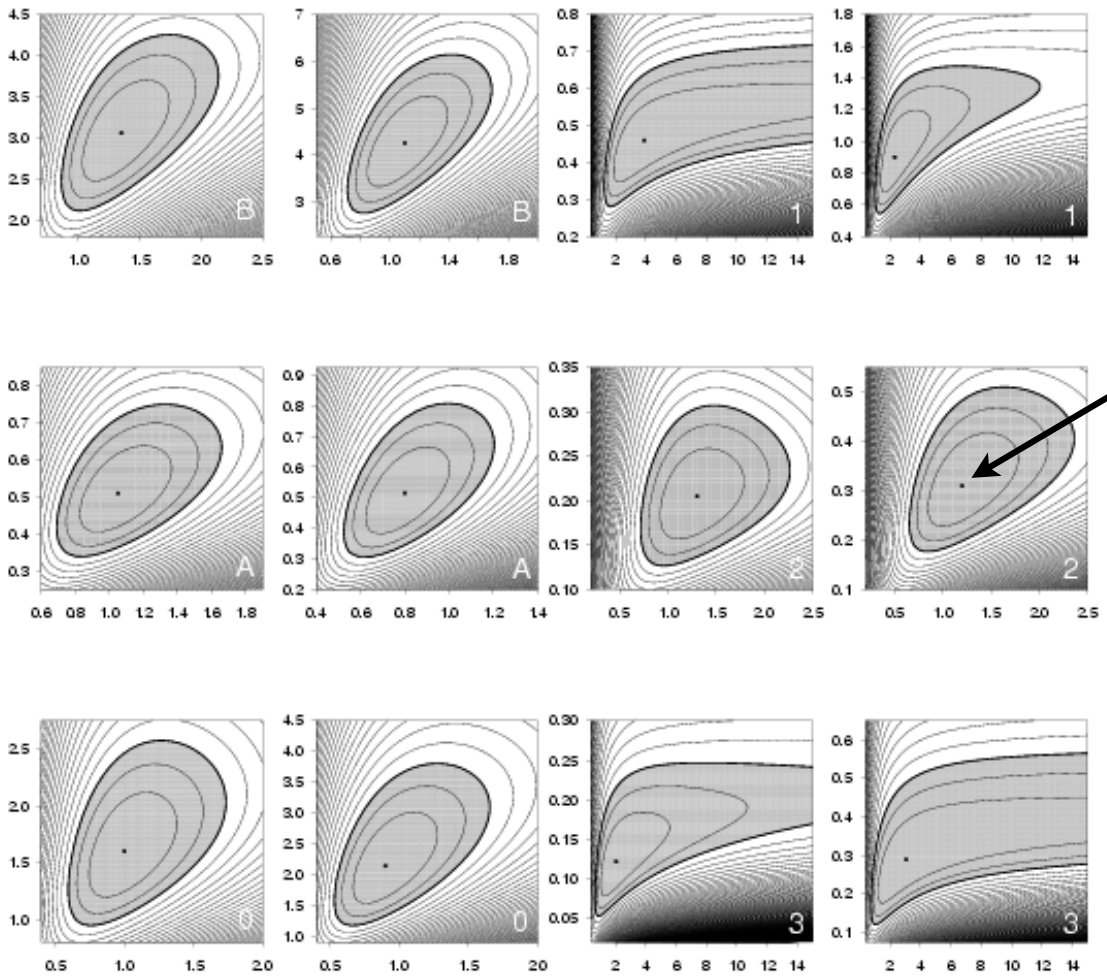


$$L(\theta|d) = \prod^{k} L(\theta|d_k)$$

Data $d = \frac{k_1}{n_1}, \frac{k_2}{n_2}, \frac{k_3}{n_3}, \frac{k_4}{n_4}, \ldots$

To minimize negative log likelihood (- ln L)

Parameter uncertainty



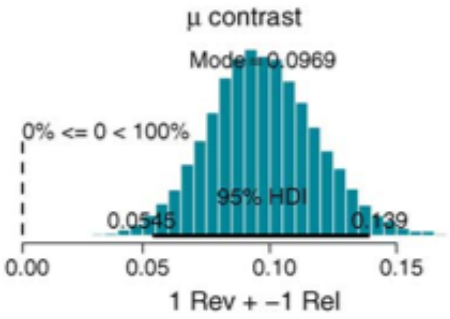
the process described on the previous slides results in a single set of parameters

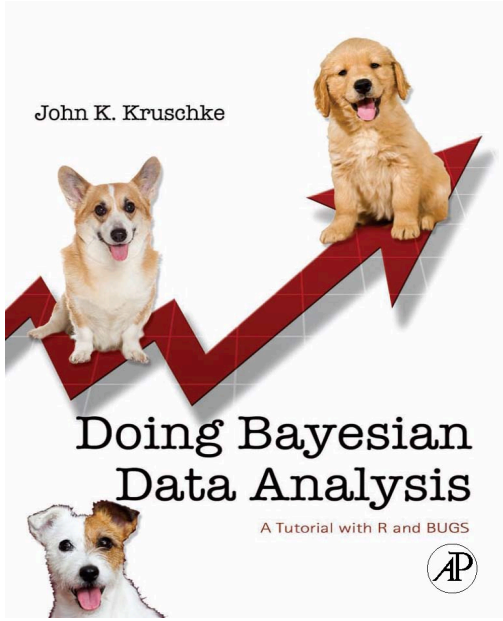
a “point” estimate.

Parameter uncertainty

- If we run the experiment again and get different data, the ML parameter will likely change
- How can we capture some of this uncertainty? In other words, when we tell people the best parameters are X, Y, Z, can we also convey our confidence in those values?

1. Be a Bayesian





Parameter uncertainty

- If we run the experiment again and get different data, the ML parameter will likely change
- How can we capture some of this uncertainty? In other words, when we tell people the best parameters are X, Y, Z, can we also convey our confidence in those values?

2. Analyze your model parameters as "measurements" across subjects

Psychological Assessment
2002, Vol. 14, No. 3, 253–262

1040-3590/02/$5.00 DOI: 10.1037//1040-3590.14.3.253

A Contribution of Cognitive Decision Models to Clinical Assessment: Decomposing Performance on the Bechara Gambling Task

Jerome R. Busemeyer and Julie C. Stout
Indiana University Bloomington

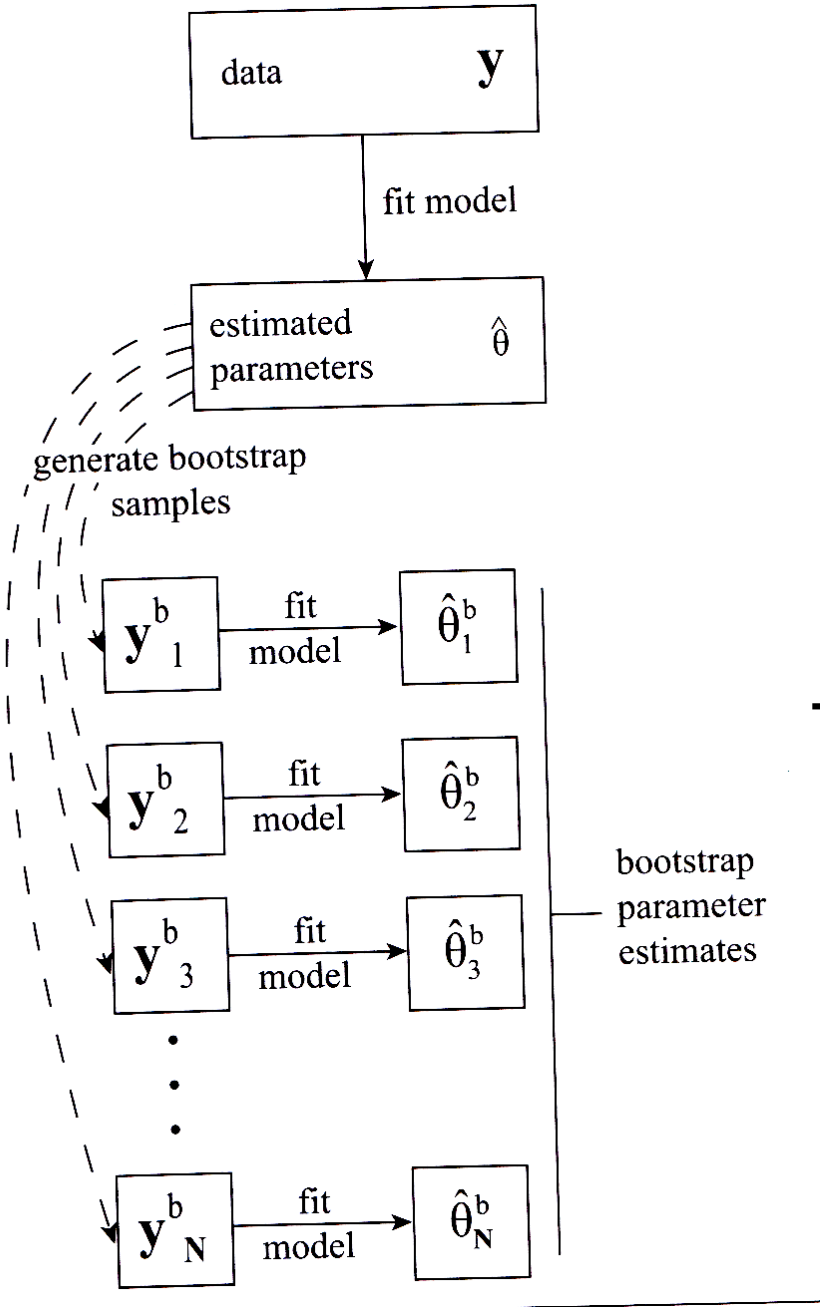
Table 2
Parameter Estimates From Expectancy–Valence Model

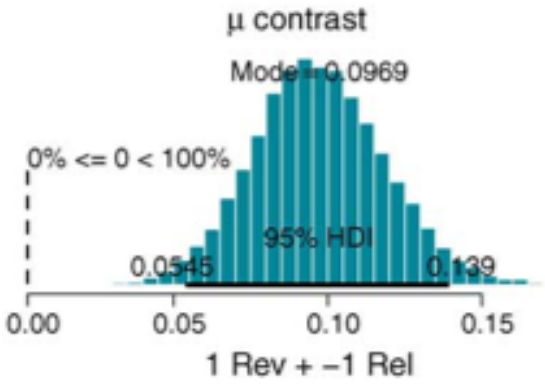
	Updating rate			Attention weight			Sensitivity		
Group	*M*	*Mdn*	*SD*	*M*	*Mdn*	*SD*	*M*	*Mdn*	*SD*
Healthy	0.34	0.16	0.39	0.35	0.32	0.30	0.32	0.68	1.58
Huntington	0.58	0.78	0.46	0.40	0.17	0.47	−0.89	−0.78	2.13
Parkinson	0.29	0.09	0.40	0.28	0.20	0.32	0.36	0.01	1.99

Parameter uncertainty

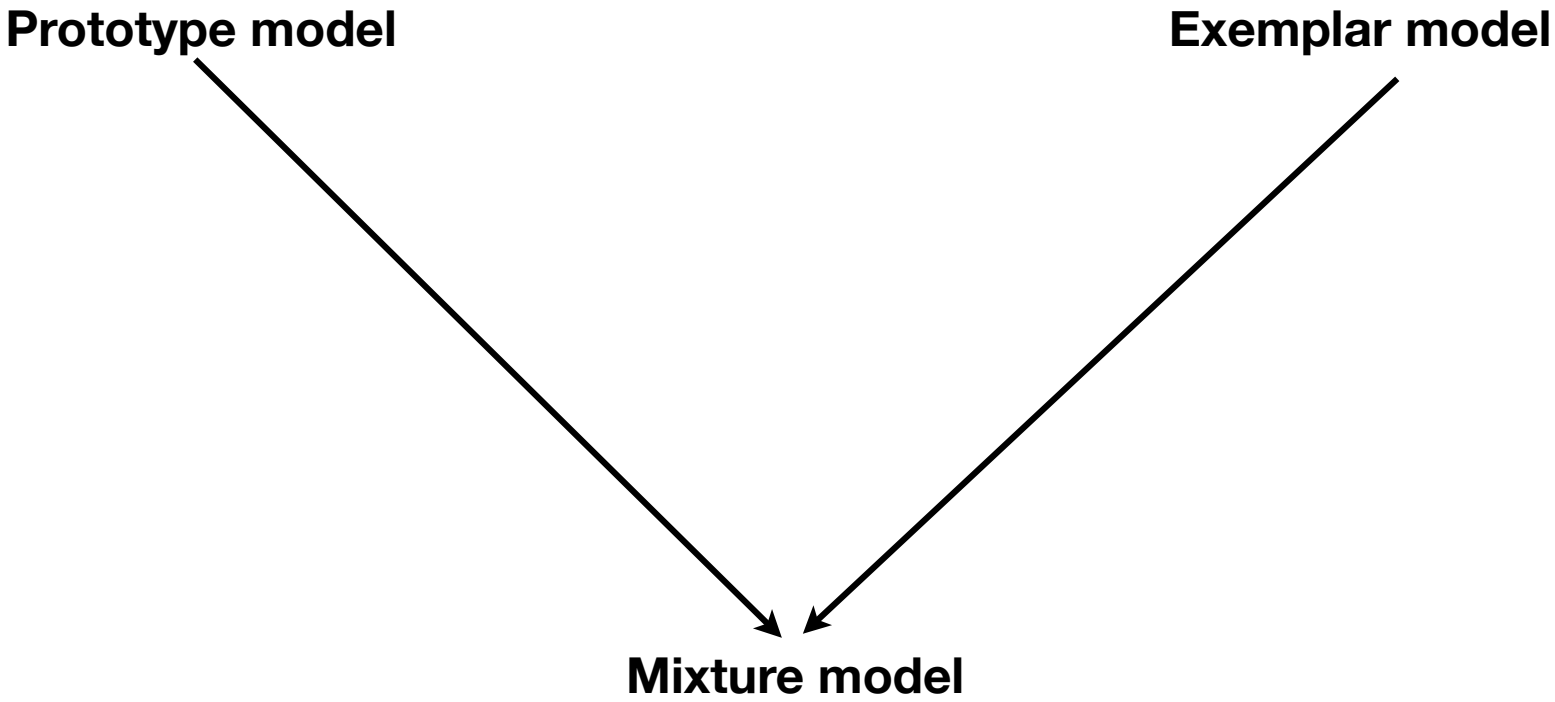
- If we run the experiment again and get different data, the ML parameter will likely change
- How can we capture some of this uncertainty? In other words, when we tell people the best parameters are X, Y, Z, can we also convey our confidence in those values?

3. Bootstrap!

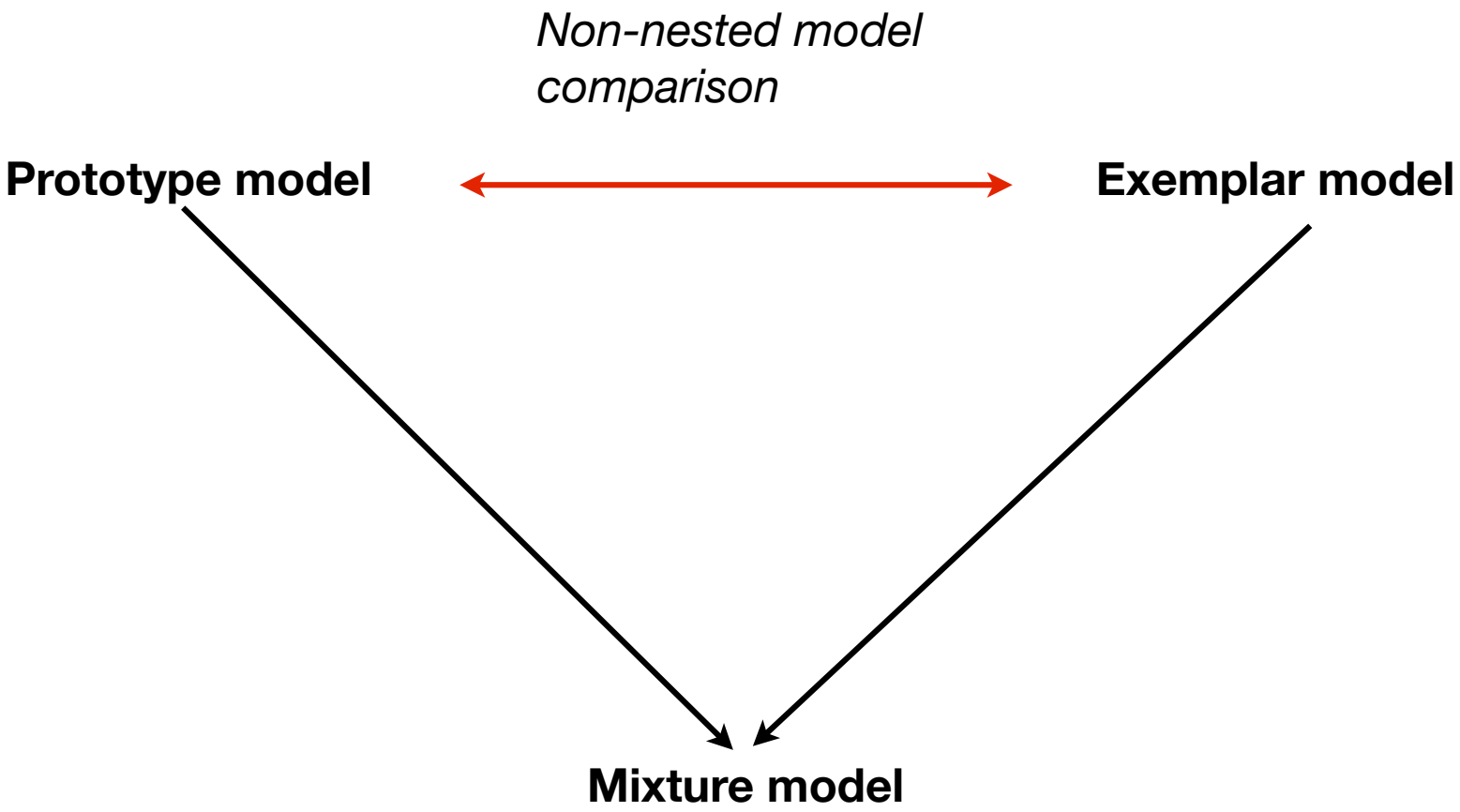

data
y
fit model
estimated parameters
θ̂
generate bootstrap samples
y b 1
fit model
θ̂ b 1
y b 2
fit model
θ̂ b 2
y b 3
fit model
θ̂ b 3
y b N
fit model
θ̂ b N
bootstrap parameter estimates


μ contrast
Mode = 0.0969
0% <= 0 < 100%
95% HDI
0.0545
0.139
0.00
0.05
0.10
0.15
1 Rev + −1 Rel

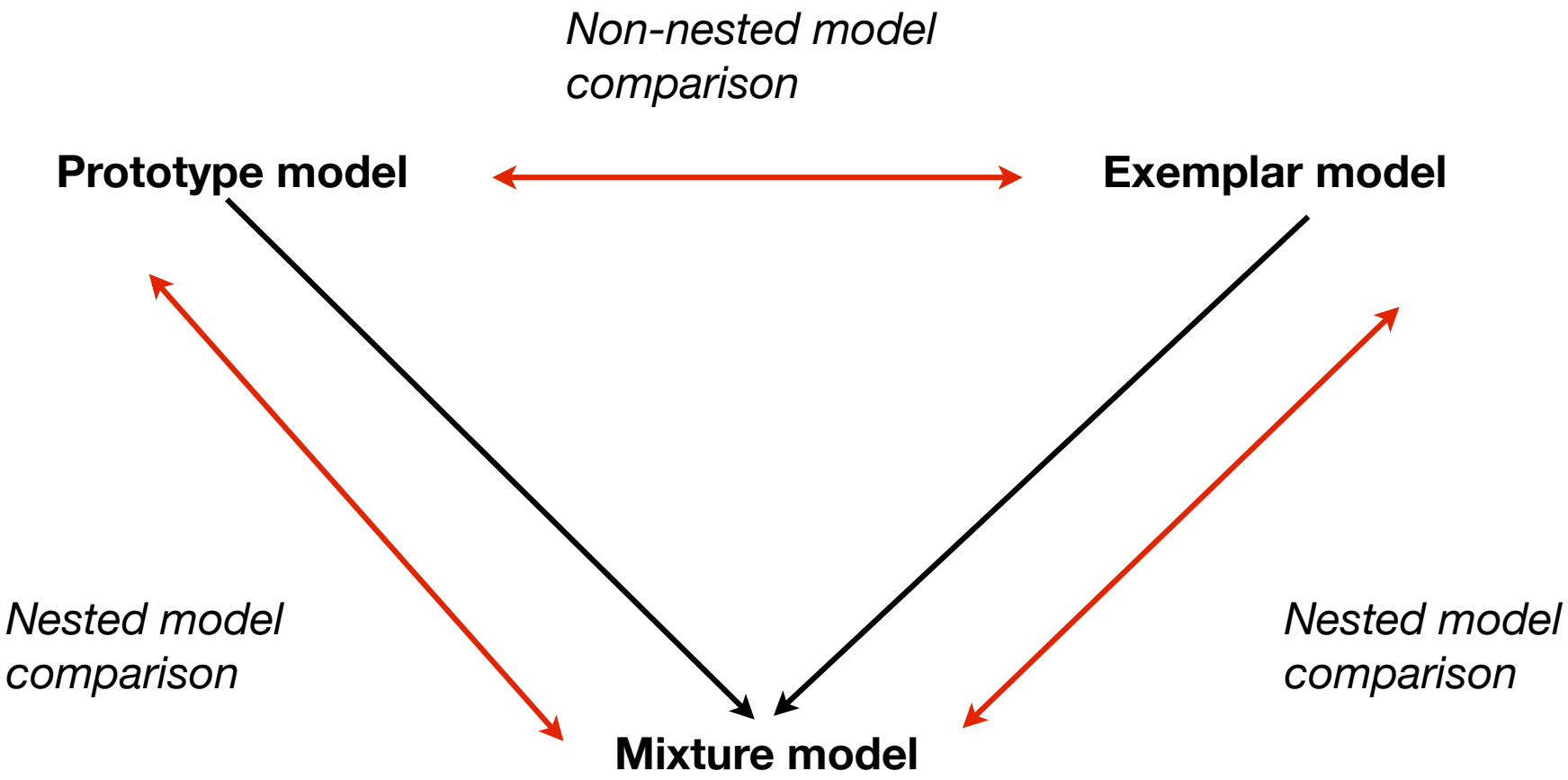
Model comparison

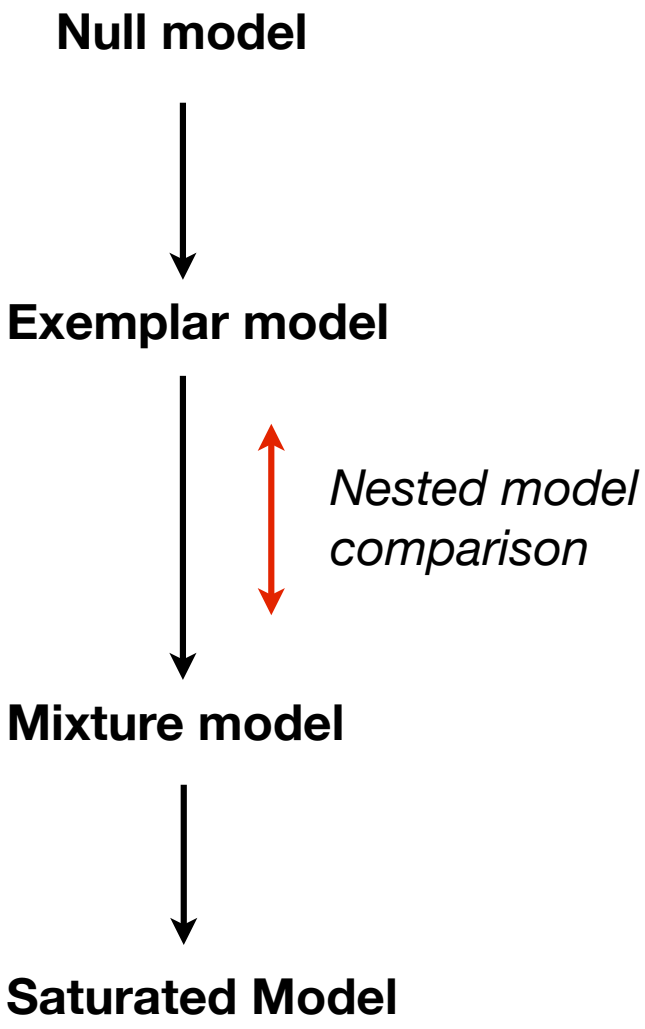


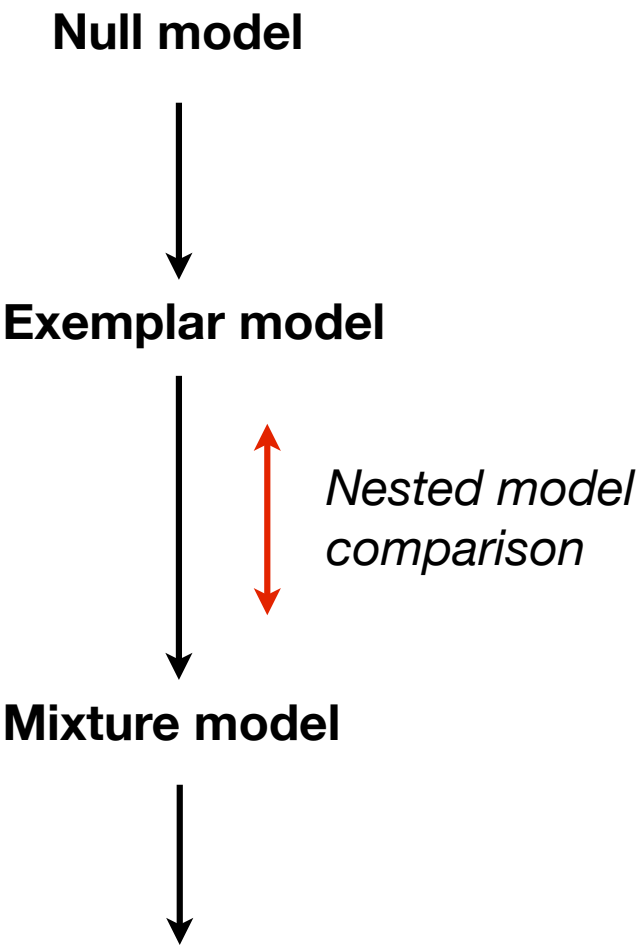
Model comparison



Model comparison



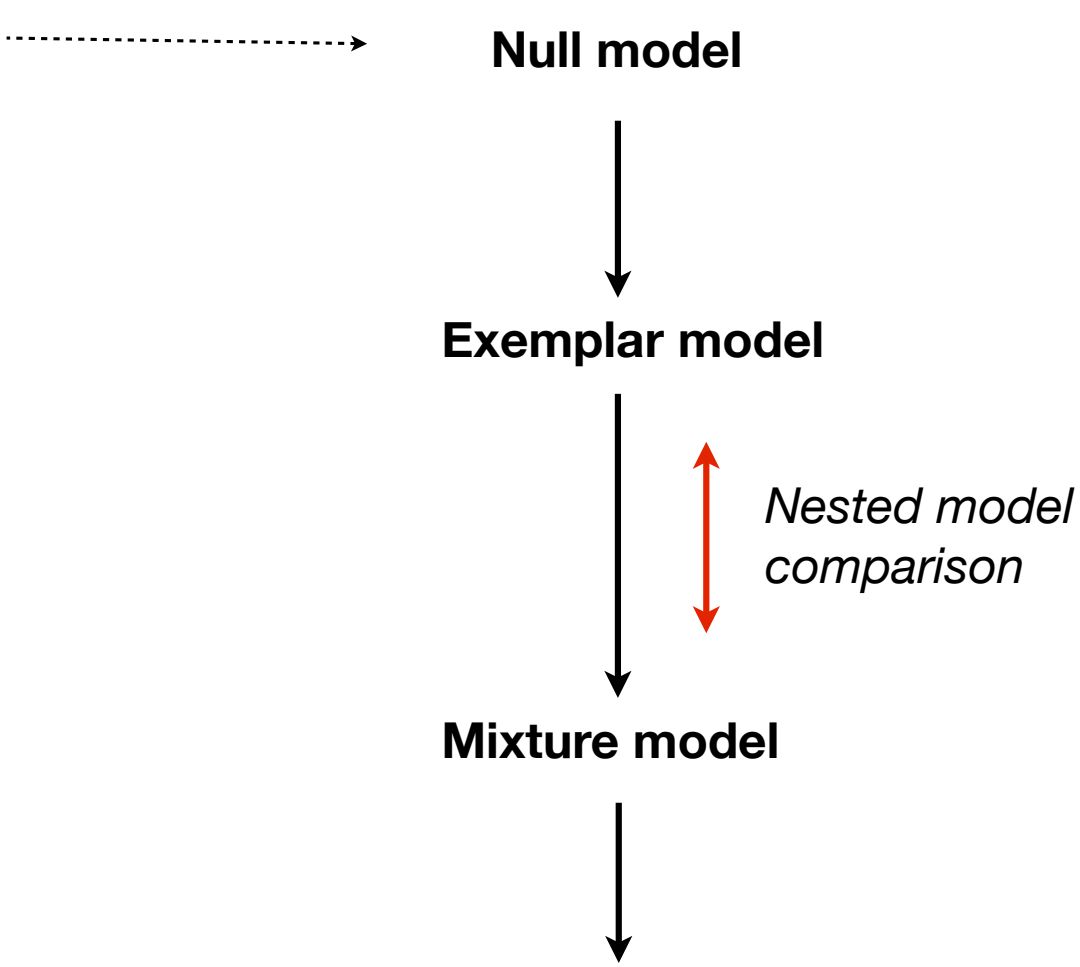

Null model
Exemplar model
Nested model comparison
Mixture model
Saturated Model



One parameter per data point

Best any model can do!

Not as bad as any model can do, but a good floor



One parameter per data point

Best any model can do!

Not as bad as any model can do, but a good floor

Null model

To be useful this model has to fit better than the null!

Exemplar model

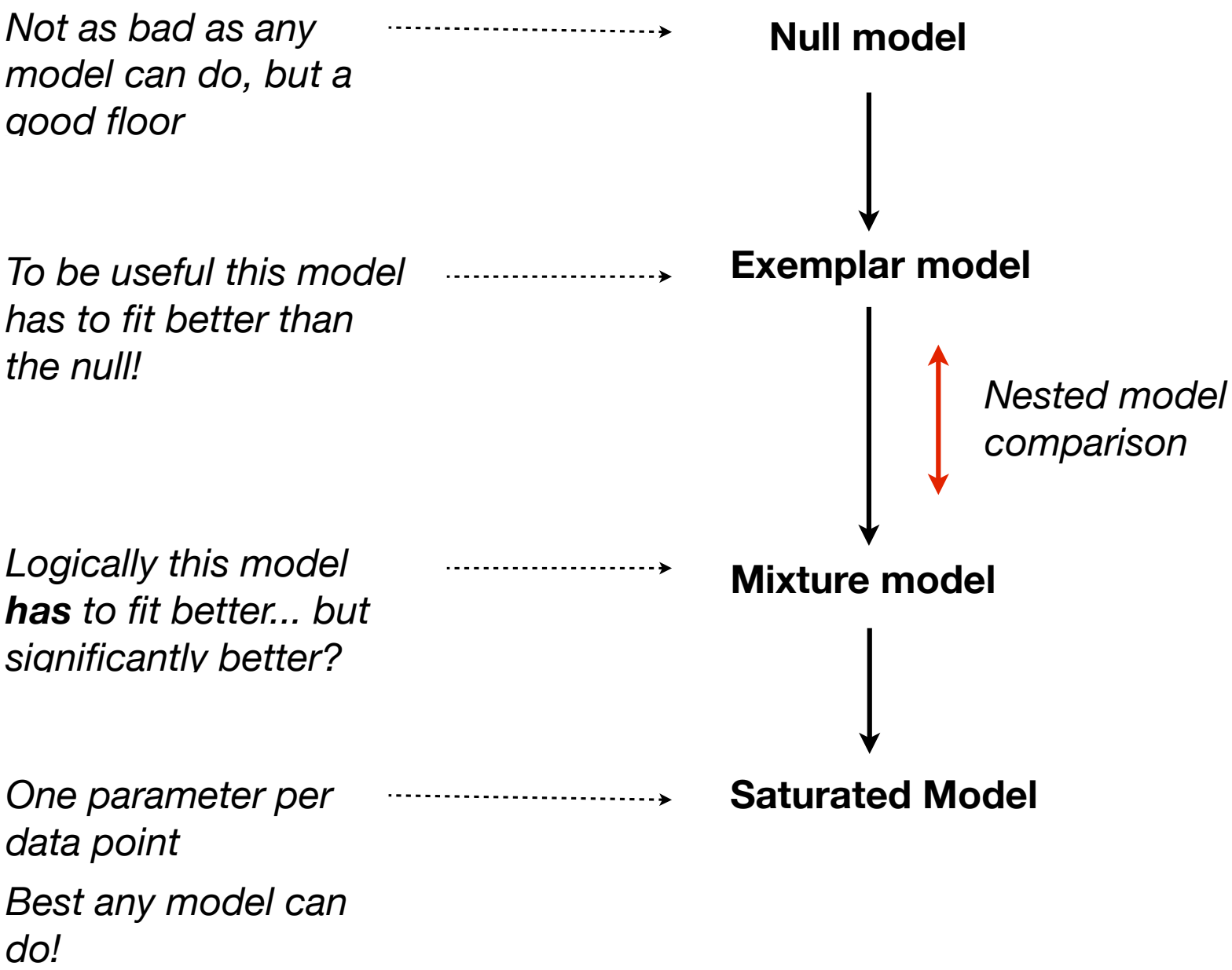
Nested model comparison

Mixture model

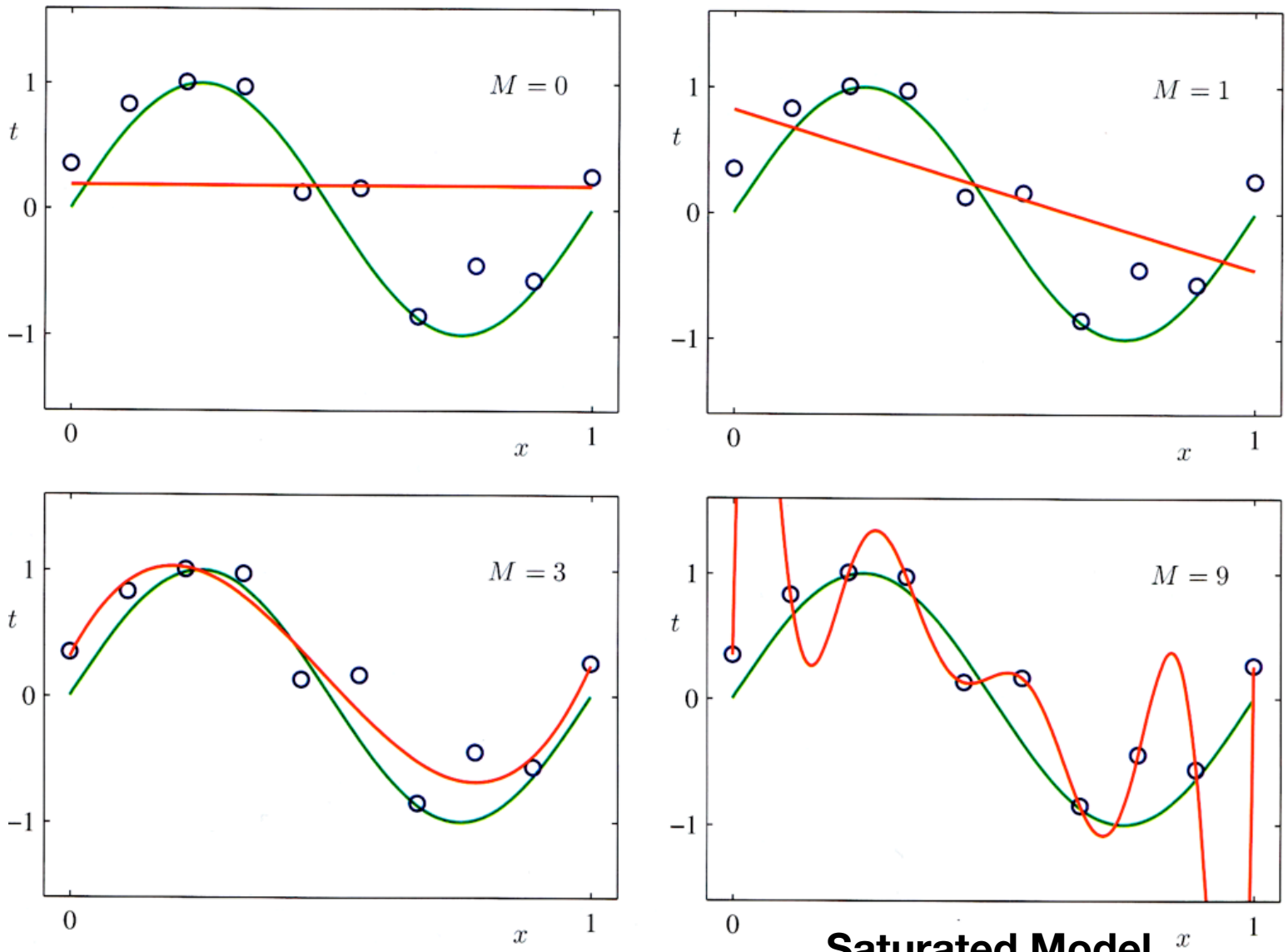
One parameter per data point

Saturated Model

Best any model can do!


Not as bad as any model can do, but a good floor
Null model
To be useful this model has to fit better than the null!
Exemplar model
Nested model comparison
Logically this model **has** to fit better... but significantly better?
Mixture model
One parameter per data point
Saturated Model
Best any model can do!

Null Model



Saturated Model

In an ideal world, you want to fit as well as the saturated model, but using no "free" parameters.

More on “nesting”

- Fundamental requirement is that one model is a reduced version of another model
- Reducing means that you clamp the value of some parameter to zero or some null value.
- **The question is if the improvement in fit provided by the extra flexibility is enough to justify the added complexity in the bigger model**

Likelihood of the model given the data **L** = p(d|m)

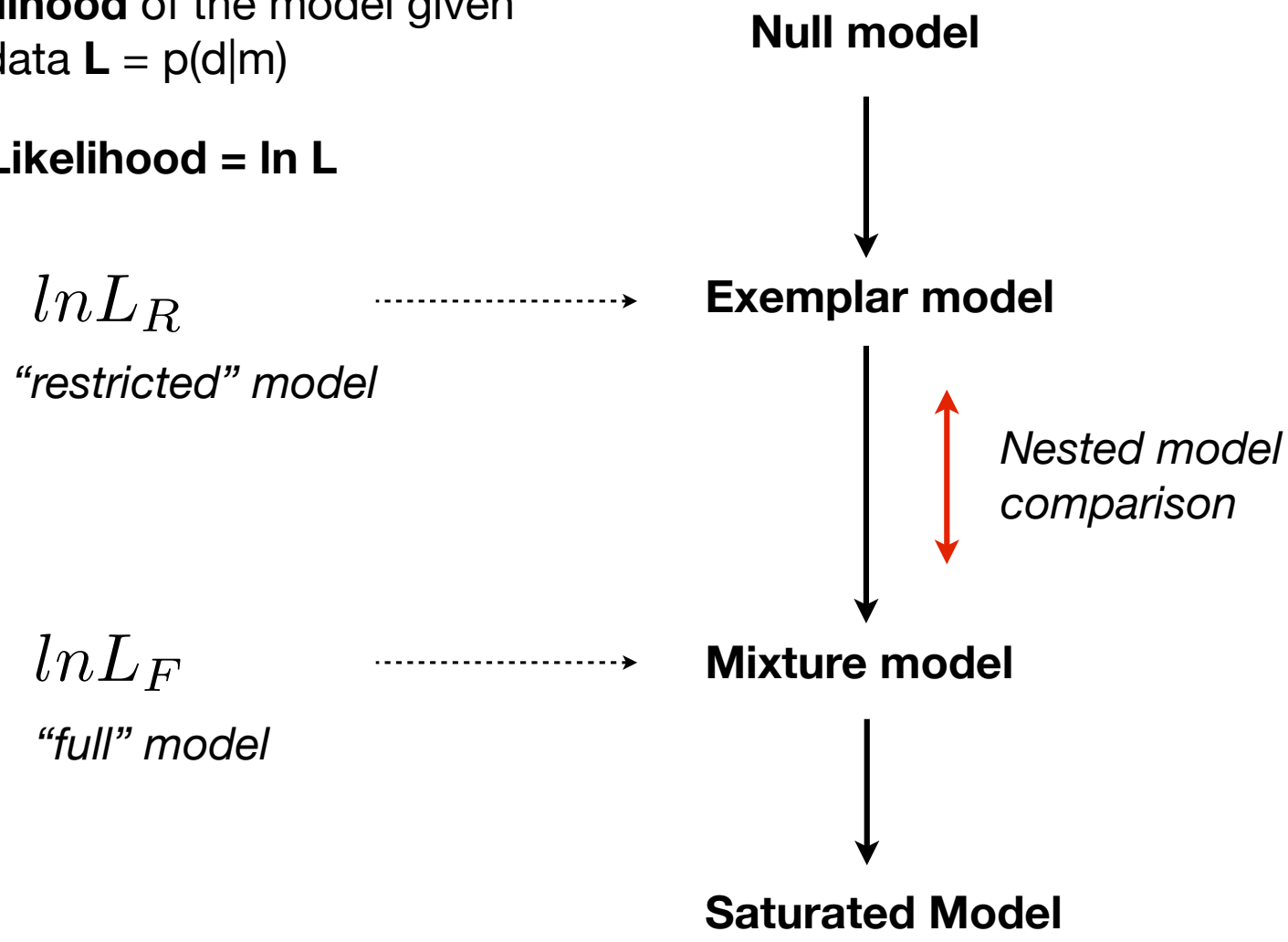
log Likelihood = ln L

lnL_R

"restricted" model

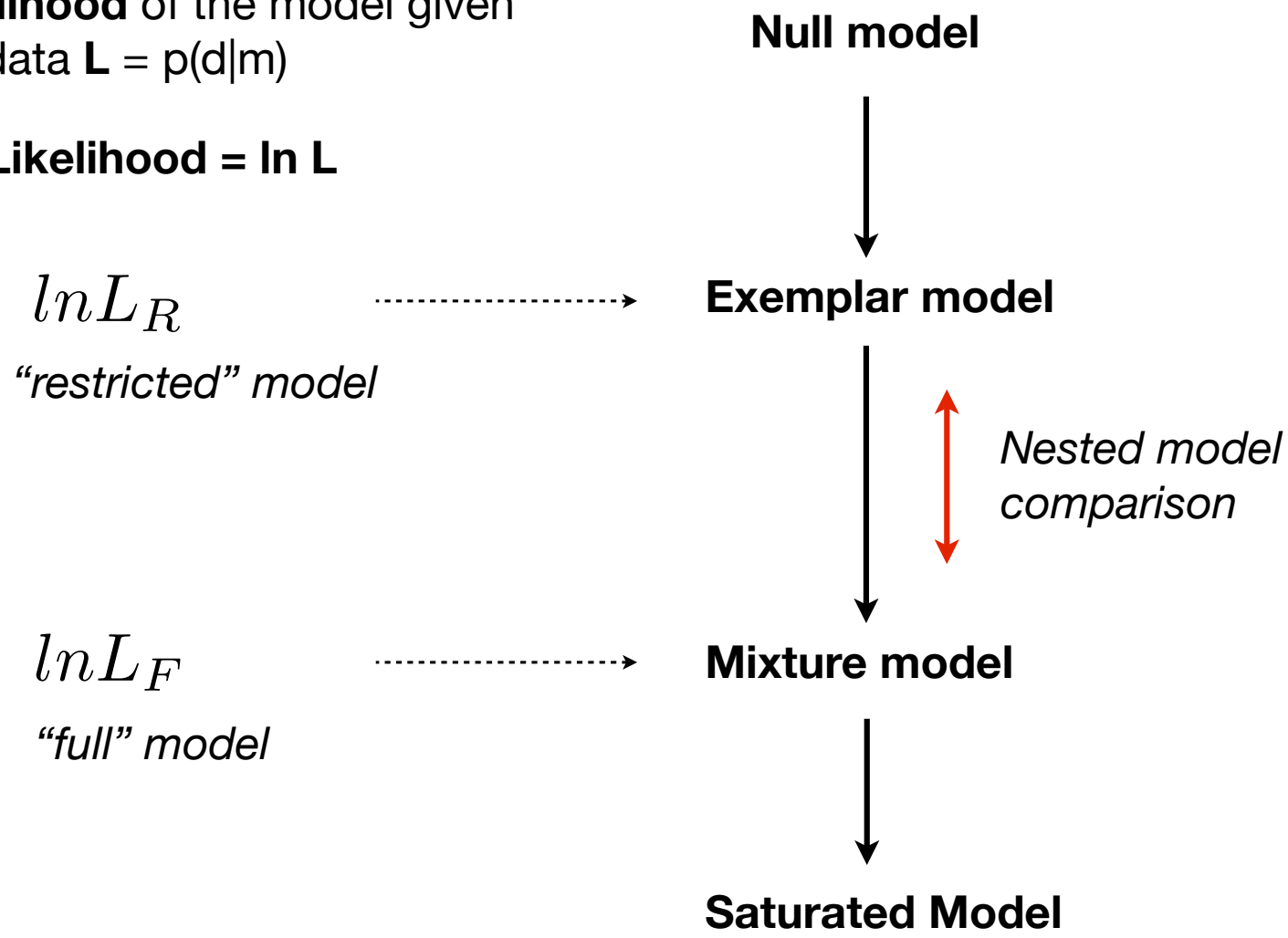
lnL_F

"full" model



Likelihood of the model given the data **L** = p(d|m)

log Likelihood = ln L



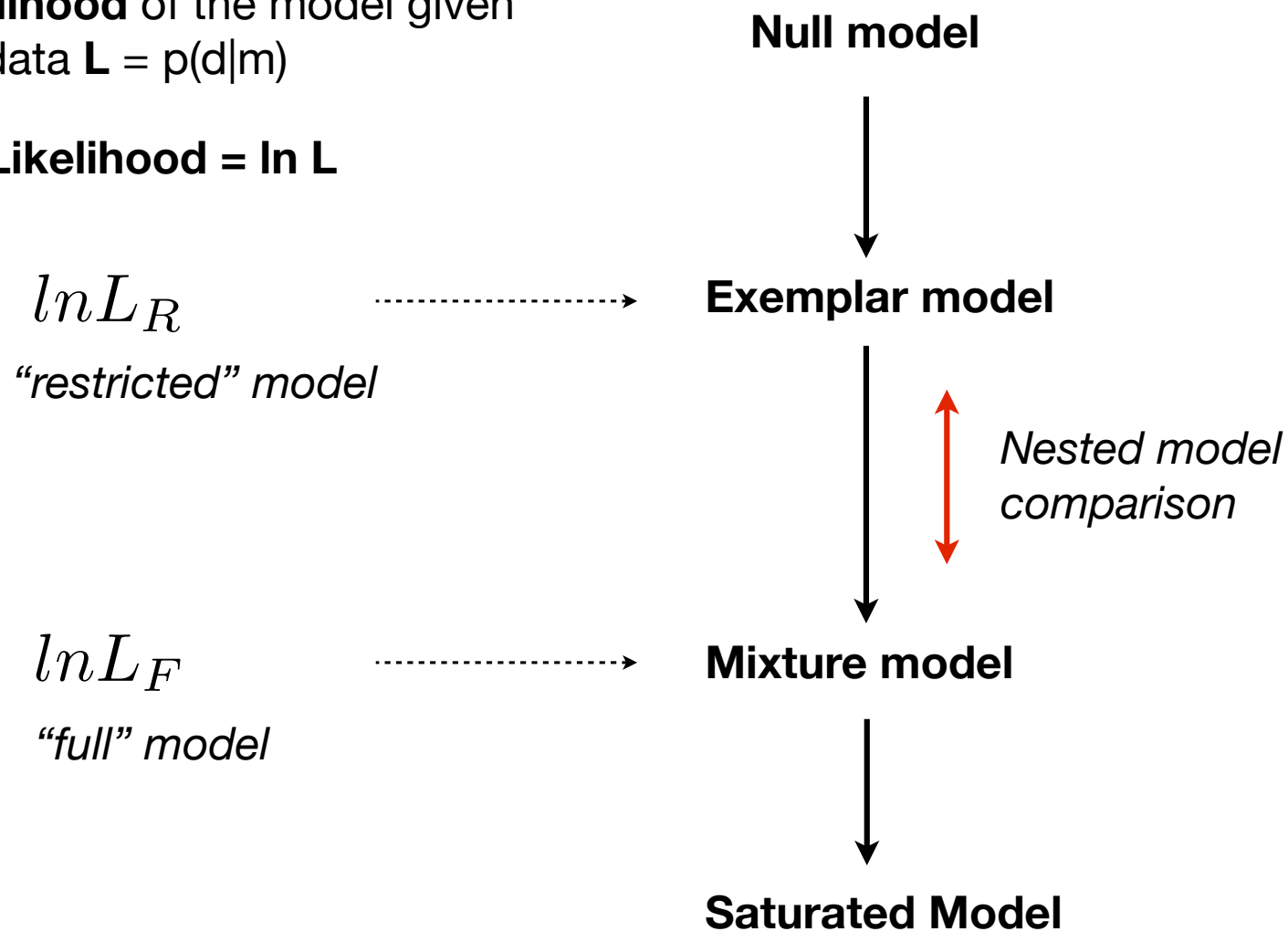
$lnL_F - lnL_R$

difference in fit

$2(lnL_F - lnL_R)$

Likelihood of the model given the data **L** = p(d|m)

log Likelihood = ln L



$$2(lnL_F - lnL_R)$$

god says. (convenient relationship between deviance and X^2)

Likelihood of the model given the data **L** = p(d|m)

log Likelihood = ln L

Null model

lnL_R ⇢ **Exemplar model**

"restricted" model

Nested model comparison

lnL_F ⇢ **Mixture model**

"full" model

Saturated Model

$G^2 = 2(lnL_F - lnL_R)$ ⇠ *distributed as* χ^2 *with NparamsFull - NparamsRest degrees of freedom*

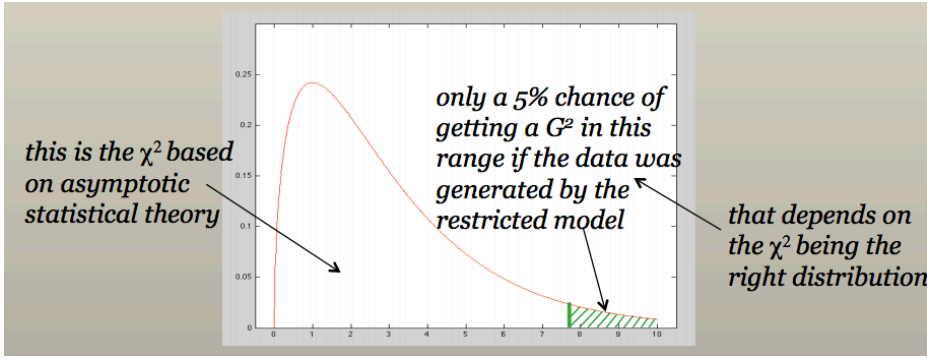
Likelihood ratio test

$$G^2 = 2(lnL_F - lnL_R)$$ ⟵

distributed as

with NparamsFull - NparamsRest degrees of freedom

If G^2 exceeds the critical value in the χ^2 distribution with the specified degrees of freedom and particular alpha-level, then the the restricted model is significantly worse than the full model.



Likelihood ratio test

$$G^2 = 2(lnL_F - lnL_R)$$ ⟵

distributed as
with NparamsFull - NparamsRest
degrees of freedom

Example

lnL_R = -293.12

lnL_F = -263.12

df = 240-45 = 195

$$G^2 = 2 \times [-263.12 - -293.82] = 61.40$$

$$X_c^2(df = 195, \alpha = .05) = 228.6$$

the restricted model is NOT significantly worse

Likelihood ratio test

$$G^2 = 2(lnL_F - lnL_R)$$

distributed as
with NparamsFull - NparamsRest
degrees of freedom

$$G^2 = 2 \times (ln\frac{L_F}{L_R})$$

Likelihood ratio test

- **Only applies to nested models**
- Null Hypothesis Testing framework - one model fits better than the other
- Alternatively, we might want to weight the evidence between to models, and to compare non-nested models.

Information criteria

- Akaike's Information Criterion (AIC)

$$AIC = -2lnL(\theta|u, M) + 2K$$

- Bayesian Information Criterion (BIC)

$$BIC = -2lnL(\theta|u, M) + KlnN$$

Information criteria

- Akaike's Information Criterion (AIC)

$$AIC = -2lnL(\theta|u, M) + 2K$$

↓ *Better fit*

↑ *More parameters*

$$AIC_{full} - AIC_{rest}$$

> 0 restricted model fit better

= 0 ... equivalence

< 0 full model fit better

Information criteria

- Akaike's Information Criterion (AIC)

$$AIC_{full} - AIC_{rest}$$

if change is in 0-2 range no support for distinguishing the models

4-7 = "considerably less" support for the model with larger AIC

>10 = no support for the larger AIC model

Information criteria

- Bayesian Information Criterion (BIC)

$$BIC = -2lnL(\theta|u, M) + KlnN$$

Related to "Bayes Factor"

$$B = \frac{p(M_1|y)}{p(M_2|y)} = exp(-\frac{1}{2}\Delta BIC)$$

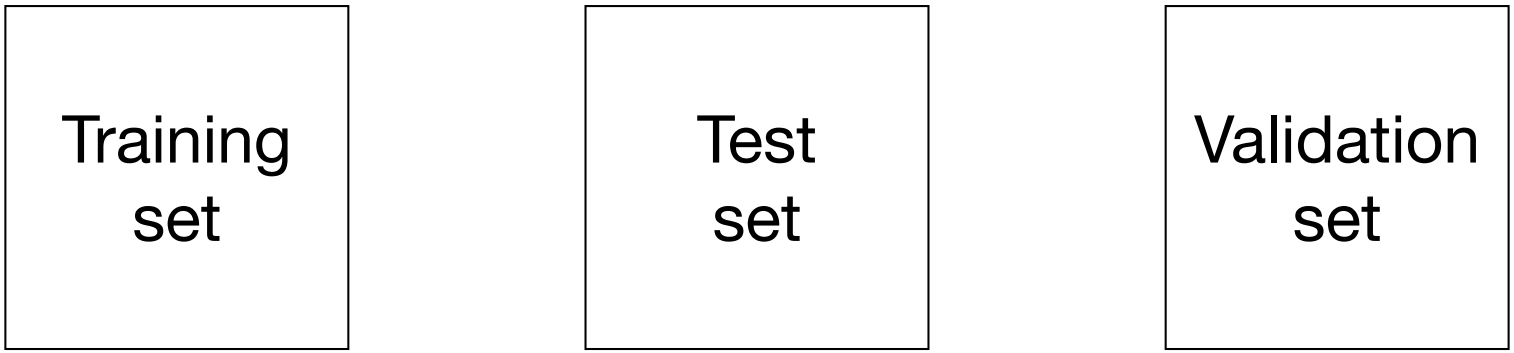
Similar "range" of values that are considered to be strong support (see Text).

Information criteria

- If you fit both AIC and BIC likely that which model is preferred can actually **conflict!**
- **Then what?!**
- BIC is generally more conservative in that it penalizes models more for extra parameters. Some people prefer it on that basis. Sometimes people report both.

Prediction!

- Measures like AIC and BIC attempt to approximate the expected log likelihood of new data
- **However,** an often better standard is literally to test your model's account of new (held out data)!



Cross validation

- Fit the model to subset of the data, then test performance on the remaining data (or fit data to one experiment, make prediction in others)
- One approach is leave-one out... fit to all the data minus 1 data point, repeat holding out a different data point each time. Compute average error on the held out point as a measure of model quality!
- “Generalization criterion methodology” (Busemeyer & Wang, 2000) - use calibration design and validation design...
    - basically fit to one part of your experiment that “calibrates the parameters” then make predictions for the interesting part
    - e.g., maybe fit model to learning trials, then evaluate on test trials

Search algorithm `fmin`

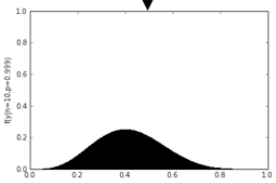
Parameters $decay$

Cognitive Model (e.g., PhonoLoop)

Model outputs $p_1, p_2, p_3, p_4, \cdots$

Likelihood model

$$P(X = k) = \binom{n}{k} p^k (1.0 - p)^{(n-k)}$$



$$L(\theta|d) = \prod^{k} L(\theta|d_k)$$

Data $d = \frac{k_1}{n_1}, \frac{k_2}{n_2}, \frac{k_3}{n_3}, \frac{k_4}{n_4}, \ldots$

To minimize negative log likelihood (- ln L)