南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Machine Learning and NeuroEngineering

机器学习与神经工程

Lecture 5 – Basic Parameter Estimation Techniques 2

Quanying Liu （刘泉影）

SUSTech, BME department

Email: liuqy@sustech.edu.cn

Lecture 4 – Recap

- Linear regression
- Discrepancy Function
  - Continuous data: Root Mean Squared Deviation (RMSD)
  - Discrete data: Chi-Squared (χ^2)
- Least-Squares Estimation （最小二乘法） to minimize the squared error
- Parameter Estimation Techniques
  - Grid search （网格搜索法）
  - Simplex （单纯形法）
  - Simulated Annealing （模拟退火算法）
- Variability in Parameter Estimates
  - Bootstrapping （自助法）

Least-square estimation – analytical solution

Squared error: $\varepsilon_i^2 = (y_i - \hat{y}_i)^2$

$$\sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1 x_i\right)^2$$

$$y = \beta_0 + \beta_1 x$$

Least Squares (L-S): Minimize squared error

Derivative of Parameters = 0

$$0 = \frac{\partial \sum \varepsilon_i^2}{\partial \beta_0} = \frac{\partial \sum \left(y_i - \beta_0 - \beta_1 x_i\right)^2}{\partial \beta_0}$$

$$= -2\left(n\bar{y} - n\beta_0 - n\beta_1 \bar{x}\right)$$

$$0 = \frac{\partial \sum \varepsilon_i^2}{\partial \beta_1} = \frac{\partial \sum \left(y_i - \beta_0 - \beta_1 x_i\right)^2}{\partial \beta_1}$$

$$= -2\sum x_i \left(y_i - \beta_0 - \beta_1 x_i\right)$$

$$= -2\sum x_i \left(y_i - \bar{y} + \beta_1 \bar{x} - \beta_1 x_i\right)$$

$$\beta_1 \sum x_i \left(x_i - \bar{x}\right) = \sum x_i \left(y_i - \bar{y}\right)$$

$$\beta_1 \sum \left(x_i - \bar{x}\right)\left(x_i - \bar{x}\right) = \sum \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

Analytical solution

$$\hat{\beta}_1 = \frac{SS_{xy}}{SS_{xx}} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

Parameter Estimation – grid search

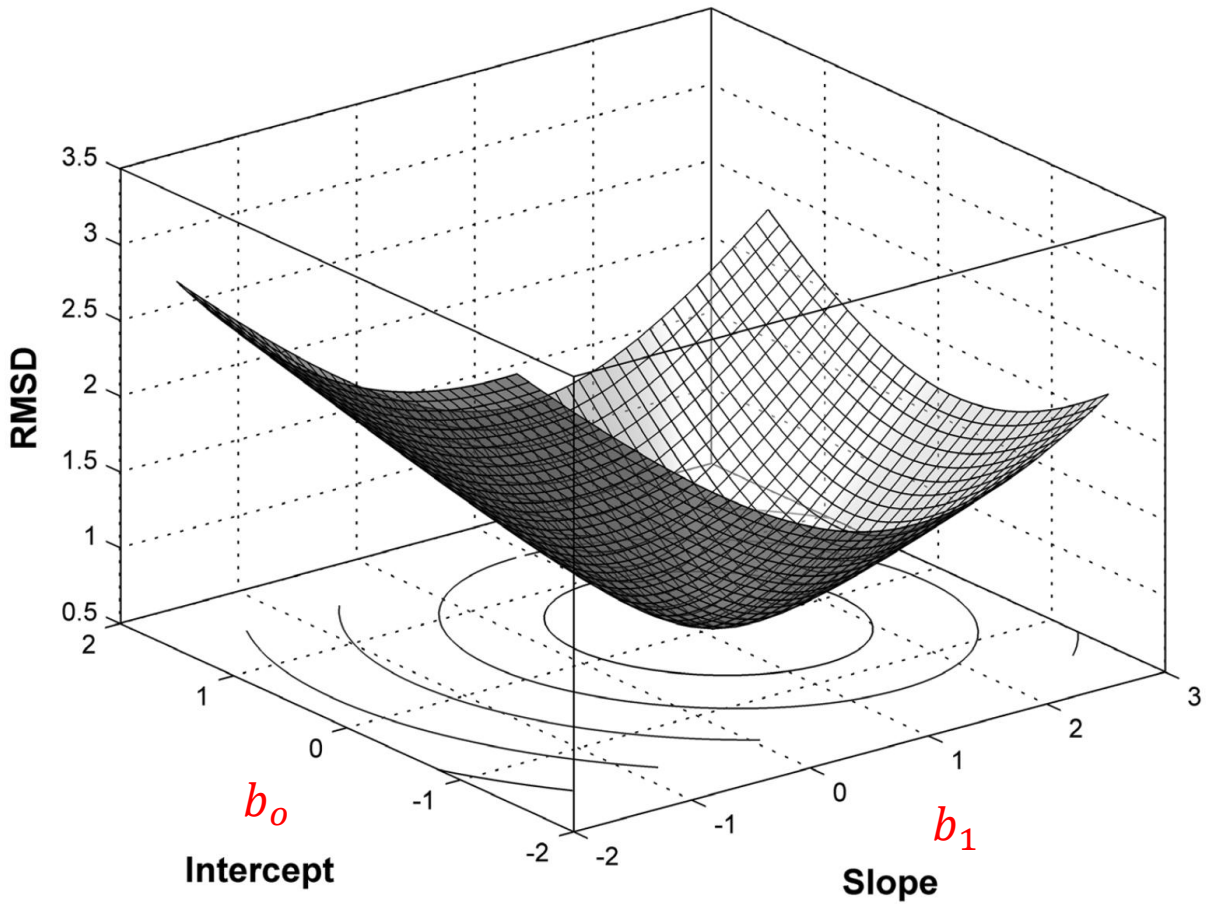
Linear regression

Model: $y = b_0 + b_1 x$

training data: (x_i, y_i) with $i = 1,2,\ldots,n$

1. Generate a grid for (b_0, b_1)
2. Calculate $\hat{y}_i = b_o + b_1 x$
3. Calculate $RMSD = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$

An "error surface"



pros: Simple, straightforward, easy to use.

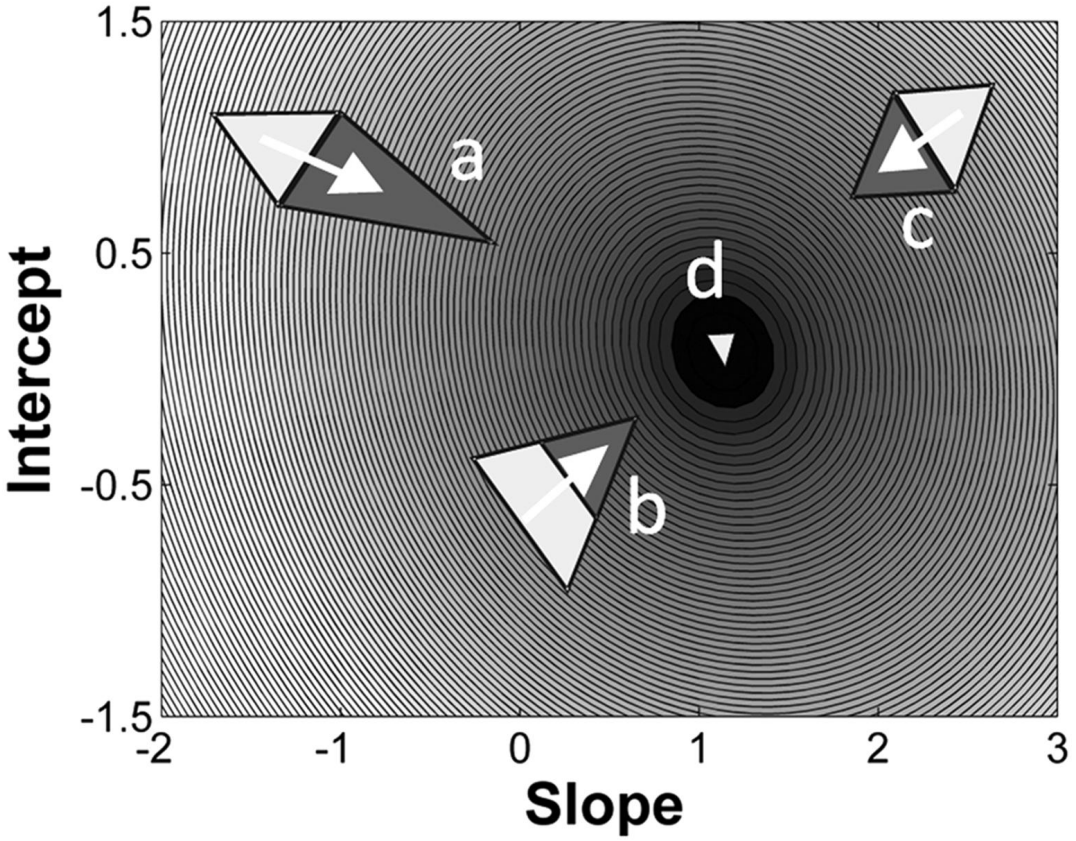
cons: Exponential increase with number of parameter

Parameter Estimation - Simplex

A **simplex** is a geometrical figure with M+1 interconnected points in M dimensions.

e.g. Simplex for Linear regression: 3 points in 2 D.

2-D projection of the error surface



Algorithm:

Create a simplex at a location given by the starting values, and calculate the *discrepancy function* for each point of the simplex.

Then, the simplex move through the parameter space:

1) be reflected/expanded: the point with the greatest discrepancy (worst fit) is flipped to the opposite side;

2) be contracted: moving the point (or points) with the worst fit closer toward the center.

Until it **converges** at the best-fitting parameter values.

Parameter Estimation - Simulated Annealing

Candidate update $\boldsymbol{\theta}_c^{(t+1)} = D(\boldsymbol{\theta}^{(t)})$

D is a “candidate function”

Difference of discrepancy value

$$\Delta f = f\left(\boldsymbol{\theta}_c^{(t+1)}\right) - f\left(\boldsymbol{\theta}^{(t)}\right)$$

Stochastic decision

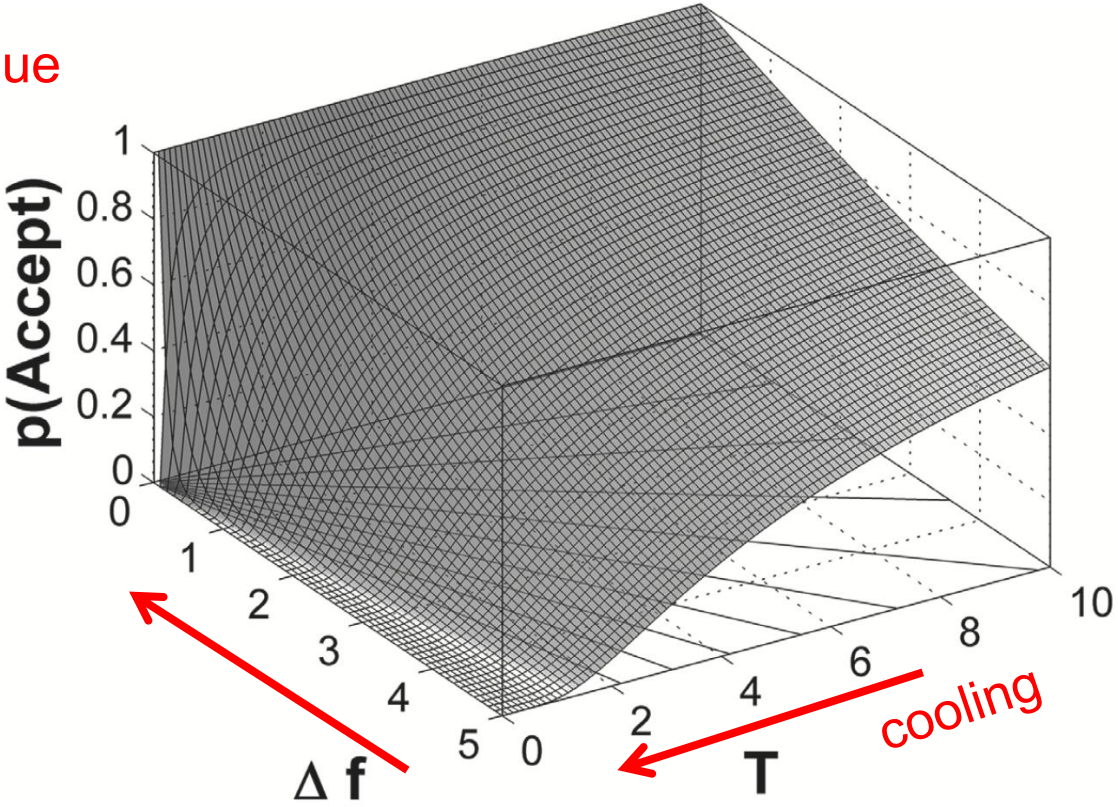
$$\boldsymbol{\theta}^{(t+1)} = \begin{cases} A(\boldsymbol{\theta}_c^{(t+1)}, \boldsymbol{\theta}^{(t)}, T^{(t)}) & \text{if } \Delta f > 0 \\ \boldsymbol{\theta}_c^{(t+1)} & \text{if } \Delta f \leq 0 \end{cases}$$

Acceptance function

$$A(\boldsymbol{\theta}_c^{(t+1)}, \boldsymbol{\theta}^{(t)}, T^{(t)}) = \begin{cases} \boldsymbol{\theta}_c^{(t+1)} & \text{if } p < e^{-\Delta f / T^{(t)}} \\ \boldsymbol{\theta}^{(t)} & \text{otherwise,} \end{cases}$$

Cooling schedule $T^{(t)} = T_0\,\alpha^t$ $T^{(t)} = T_0 - \eta\, t$

Interactions between Δf and T



Parameter Estimation - optim function in R

optim is a general-purpose optimization function.

Input:

1. Starting values of the model parameters,
2. A *function* to be minimized # such as minimize a discrepancy function
3. The method to be used (optional):

```
method = c("Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN", "Brent")
```

Output:

the best-fitting parameter estimates (a list structure)
discrepancy

Parameter Estimation - optim function in R

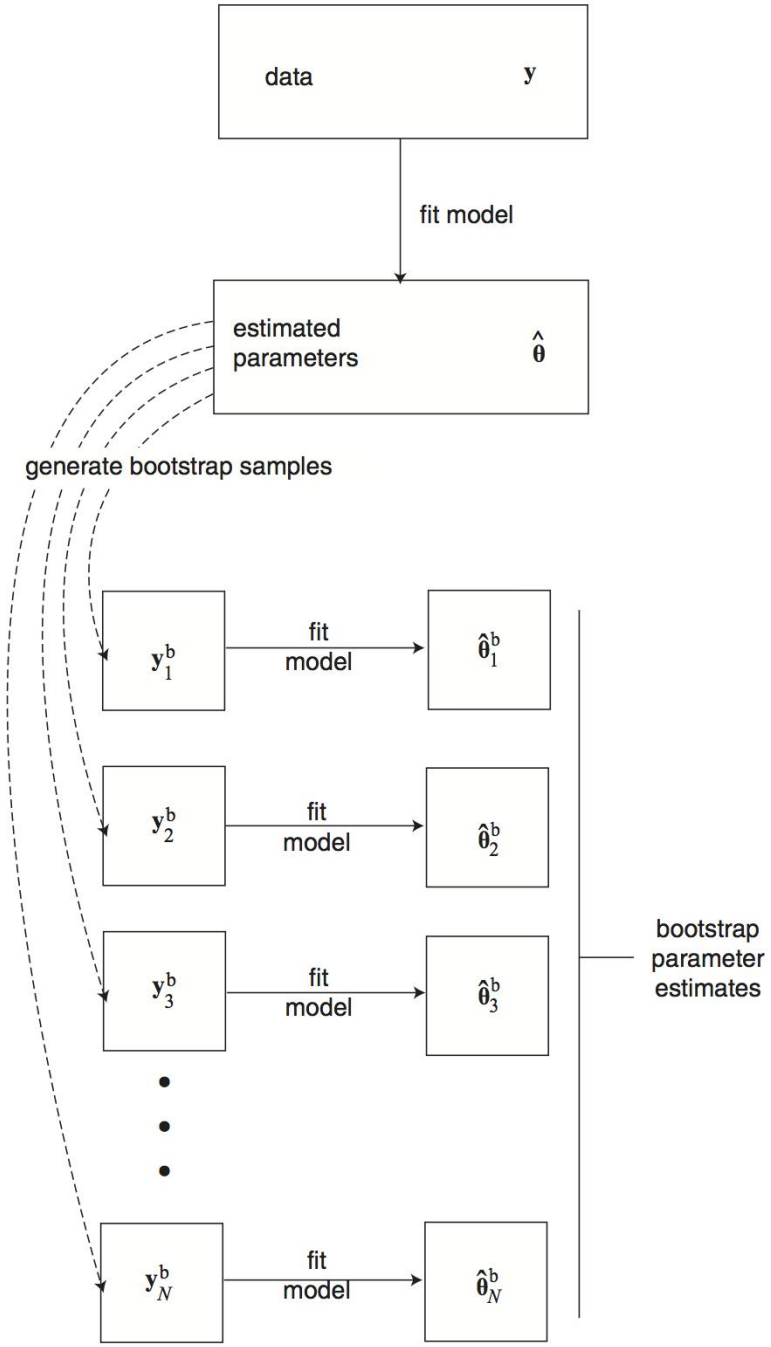
```
#plot data and current predictions
getregpred <- function(parms, data) {
  getregpred <- parms[“b0”] + parms[“b1”]*data[ ,2]          # prediction
}

#obtain current predictions and compute discrepancy
rmsd <- function(parms, data1) {
  preds <- getregpred(parms, data1)  # parms["b0"] + parms["b1"]*data[ ,2]
  rmsd  <- sqrt(sum((preds-data1[ ,1])^2)/length(preds))  # calculate RMSD
}

#assign starting values
startParms <- c(-1., .2)
names(startParms) <- c("b1", "b0")

#obtain parameter estimates
xout <- optim(startParms, rmsd, data1=data)
```

$$RMSD = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}}$$



Bootstrapping

Procedure:

We estimate parameters by fitting the model.

We generate *T* samples by running *T* simulations from the model using the estimated parameters.

(Each generated sample should contain *N* data points, where *N* is the number of data points in the original sample.)

We then fit the model to each of the *T* generated samples.

The variability across the *T* samples in the parameter estimates then gives us some idea about the variability in the parameters.

Lecture 5 – Maximum Likelihood Estimation

- Motivations for MLE
- Basics of Probabilities: properties of probability; probability functions
- What is a likelihood
- Defining a Probability Distribution
  - Specified by the NeuroPsychological Model
  - Specified by by Data Models (binomial distribution)
- Finding the Maximum Likelihood

Motivations for MLE

In Lecture 4, we have described the basics of parameter estimation by minimizing the discrepancy (loss function) between the data and the model's predictions.

What are the problems of **least-square estimation** (LSE)?

- no statistical insights
- cannot show the contribution of new evidence

Maximum likelihood estimation is deeply rooted in statistical theory.

- more accurate on average with increasing sample size
- the relative weight of evidence for a particular hypothesis
- …

Basics of Probabilities

- **Probability**
  - **Samples**: throwing dice (random experiment) generates new samples
  - **Outcomes**: what we get from the sampling process
  - **Sample space Ω**: the set of all possible outcomes of the experimen
  - **Event**: a sub-set of the sample space

Examples:

1) A toss of a coin: $\Omega = \{\text{h,t}\}$.
2) Two successive tosses of a coin:
3) A toss of two dice: $\Omega = \{(i,j) :$
4) The measurement of a length
   where $\mathbf{R}_+$ denotes the positive
   result of the measurement, and
5) The lifetime of a light-bulb: Ω

- the *contrary* event is interpreted as the complement set A^c;
- the event "A **or** B" is interpreted as the union $A \cup B$;
- the event "A **and** B" is interpreted as the intersection $A \cap B$;
- the *sure* event is Ω;
- the *impossible* event is the empty set $\emptyset$;
- an **elementary event** is a "singleton", i.e. a subset $\{\omega\}$ containing a single outcome ω of Ω.

Probability Essential, by Jean Jacod and Philip Protter

Basics of Probabilities

- **Define the Probability**

With each event A, one associates a number denoted by **P(A)** and called the "probability of A".

This number measures the likelihood of the event A to be realized a priori, before performing the experiment. It is chosen between 0 and 1, and the more likely the event is, the closer to 1 this number is.

1. $0 \leq P(A) \leq 1$,
2. $P(\Omega) = 1$,
3. $P(A \cup B) = P(A) + P(B)$ if $A \cap B = \emptyset$.

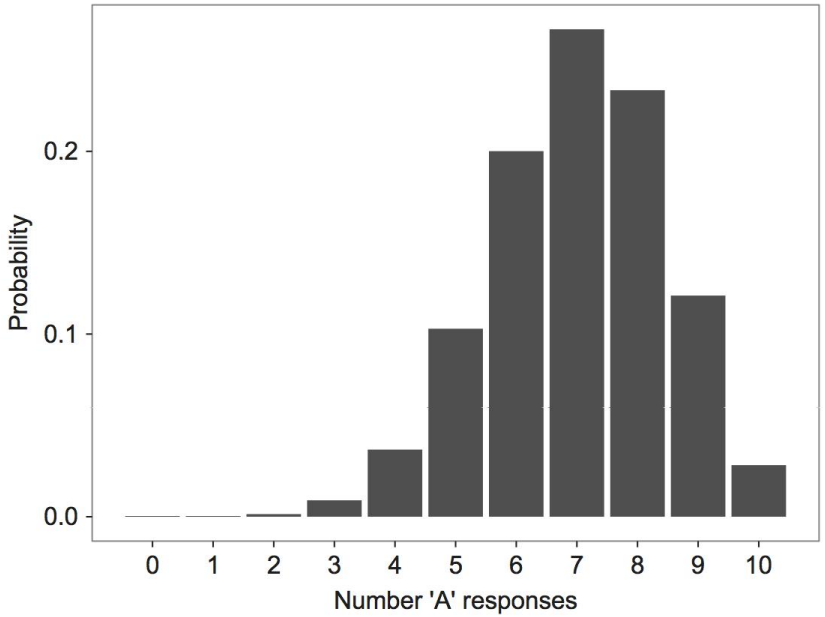
Probability Essential, by Jean Jacod and Philip Protter

Basics of Probabilities

- **Joint probability**
  - ***P*(*A*, *B*)**, which gives the probability that both *A* and *B* occur
- **Conditional probability**
  - ***P*(*A* | *B*)**, which gives the probability of observing event A given that we have observed event B.
  - $P(A, B) = P(A \mid B) \times P(B) = P(B \mid A) \times P(A)$
  - $P(data \mid model)$
  - $P(model \mid data)$
- **Independence**
  - $P(A, B) = P(A) \times P(B)$, if A and B are independent.
  - $P(A \mid B) = P(A)$

Probability functions

- A **probability function** describe the probability of all possible events the model could predict.



- Discrete events
  - ➢Probability mass function (PMF)

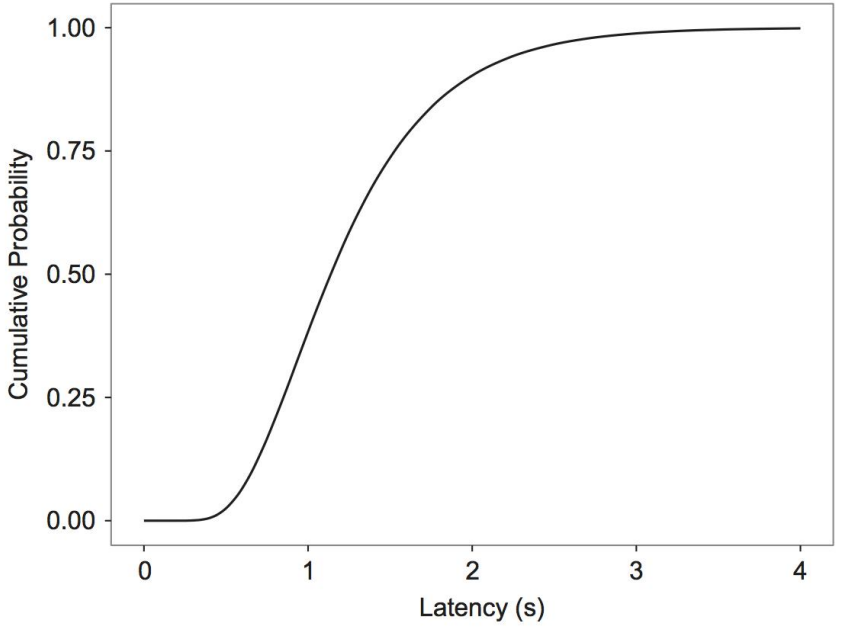
- Continuous events:
  - ➢Cumulative distribution function(CDF)
  - ➢Probability density function (PDF)

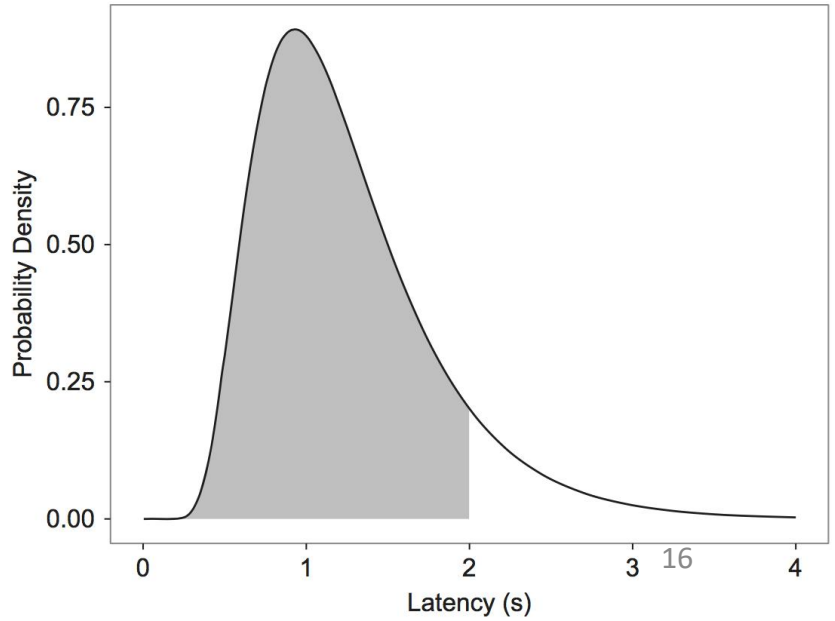
Probability functions

- A **probability function** describe the probability of all possible events the model could predict.

- Discrete events
  - ➢Probability mass function (PMF)

- Continuous events:
  - ➢Cumulative distribution function(CDF)
  - ➢Probability density function (PDF)





Probability functions

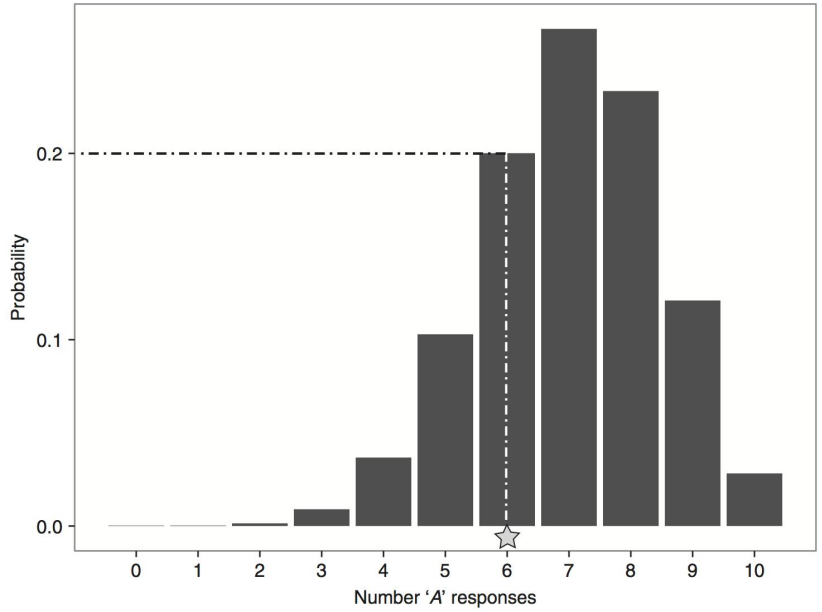
Probability function:

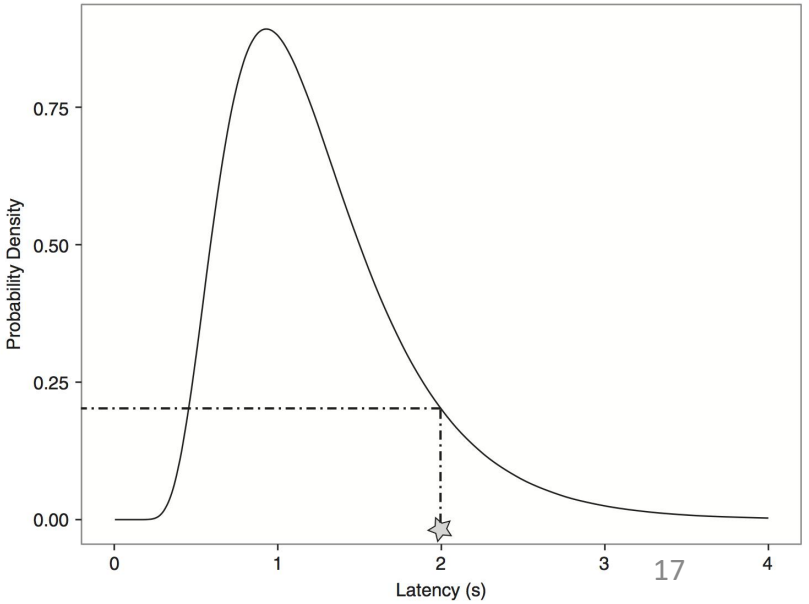
$$f(y|\boldsymbol{\theta}, M)$$

$\boldsymbol{\theta}$: a vector of parameters' values
M: a particular model
y: the data

A joint probability or probability density for the data in a data vector **y**

$$f(\mathbf{y}|\boldsymbol{\theta}) = \prod^{k} f(y_k|\boldsymbol{\theta})$$





What is a likelihood?

p(t | m)

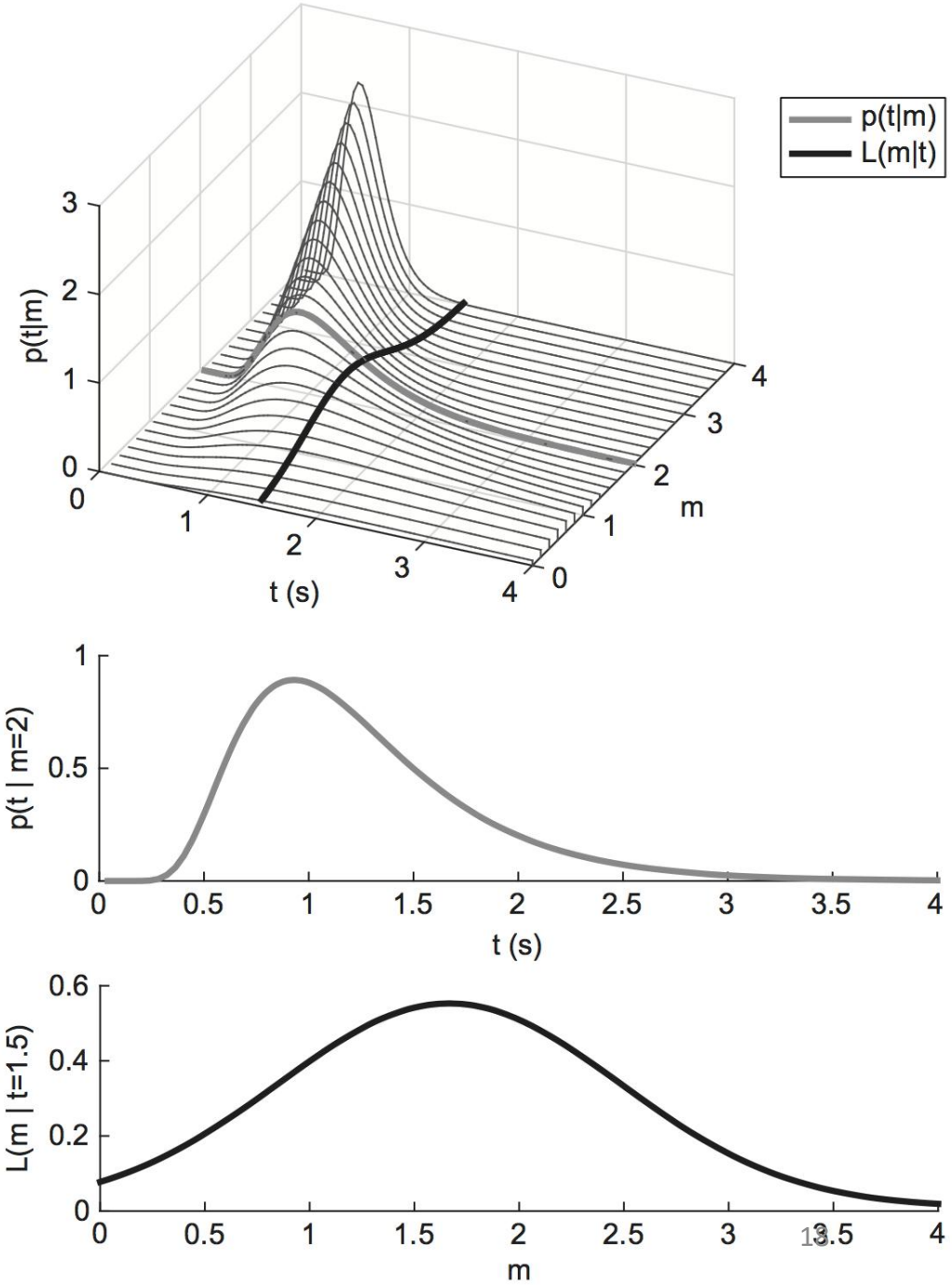
for all the possible response time t and Wald model parameter m.

p(t | m=2)

Probability distribution for response time, given m=2.

Likelihood: *L(m | t=1.5)*

Probability distribution of the model parameter m, given the observed response time t=1.5s.



What is a likelihood?

$p(N_A \mid P_A)$

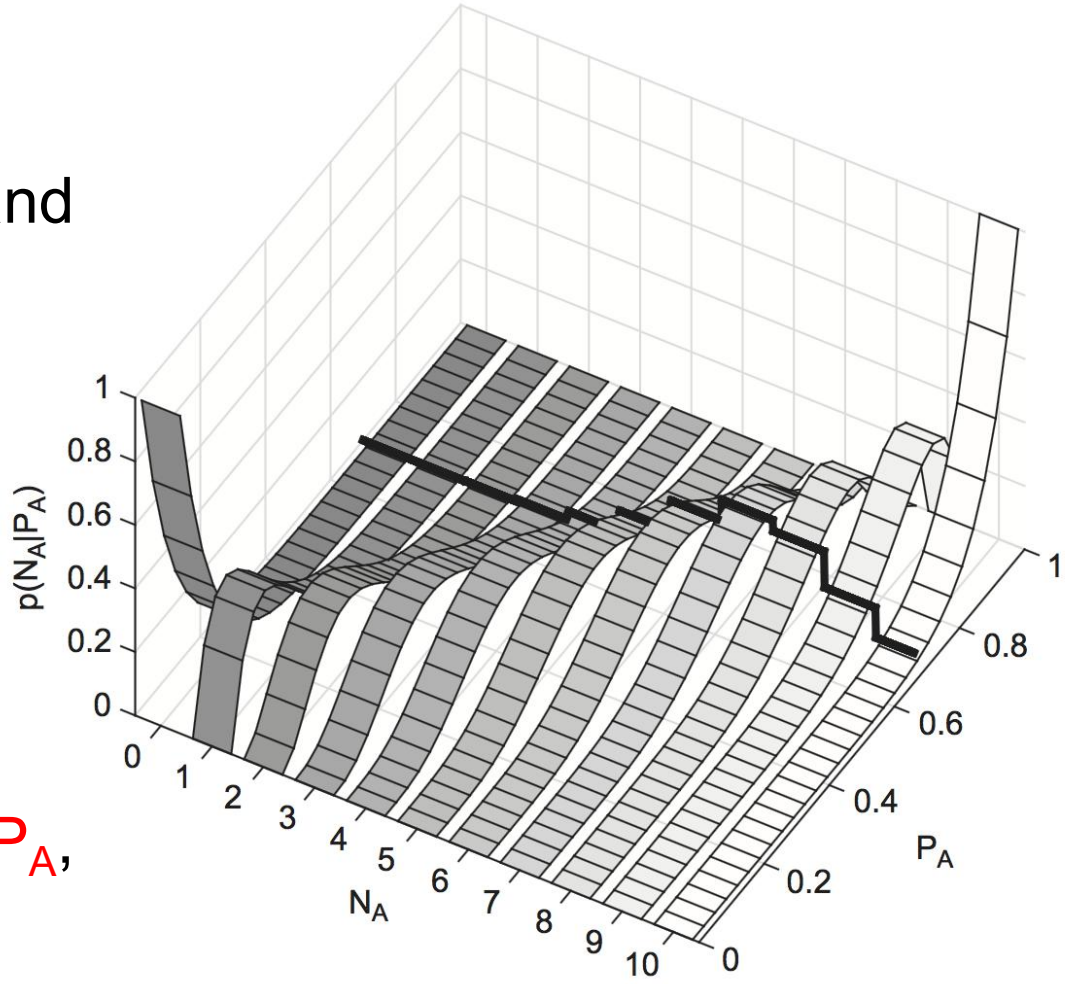
for all the possible numbers of A response N_A and binomial model parameter P_A.

$p(t \mid P_A=0.7)$

Probability distribution for N_A, given $P_A=0.7$.

$L(P_A \mid N_A=6)$

Probability distribution of the model parameter P_A, given the observed data $N_A=6$.



Defining a Probability Distribution

The **probability function** maps parameter(s) into a probability or probability density for every possible outcome (i.e., every possible data value).

- Specified by the NeuroPsychological Model (Wald distribution)
- Specified by by Data Models (binomial distribution)

Probability Function Specified by the Psychological Model

A **Wald distribution** for modelling response time, proposed by Wald in 1947.

The **shifted Wald probability density function** for the response time t:

$$f(t|a,m,T) = \frac{a}{\sqrt{2\pi\,(t-T)^3}} \exp\left(-\frac{[a-m\,(t-T)]^2}{2\,(t-T)}\right), t > T.$$

Model parameters

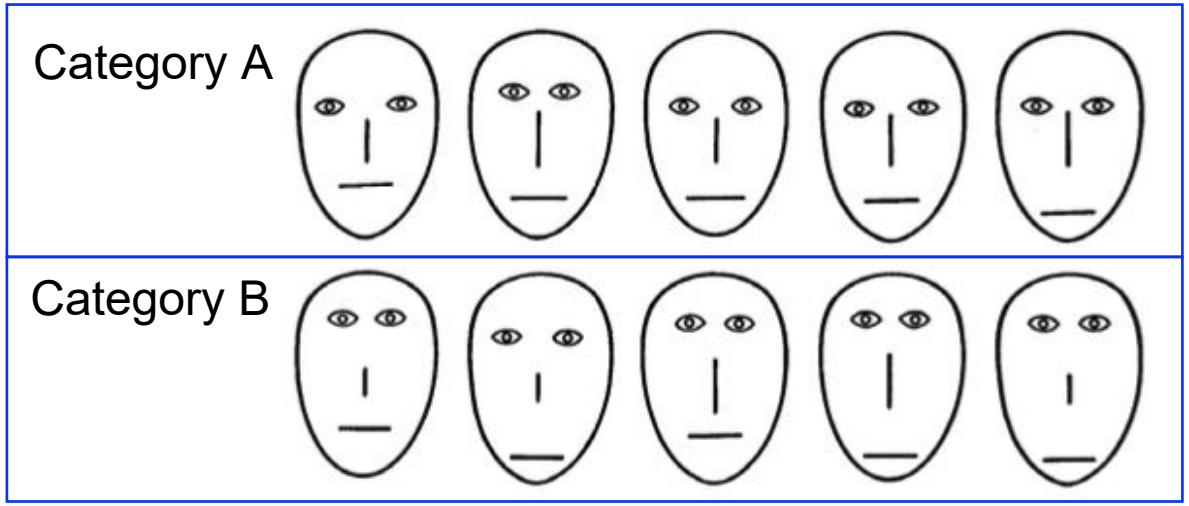
m: drift

a: the position of the response boundary

T: the added non-decision time

```
# shifted Wald probability density function
rswald <- function(t, a, m, Ter){
  ans <- a/sqrt(2*pi*(t-Ter)^3)*
           exp(-(a-m*(t-Ter))^2/(2*(t-Ter)))
}
```

Probability Function Specified by the Data model



The **distance** between two faces (i&j):

$$d_{ij} = \left(\sum_{k=1}^{K} |x_{ik} - x_{jk}|^2 \right)^{\frac{1}{2}}$$

The **similarity** between two faces:

$$s_{ij} = exp(-c \cdot d_{ij})$$

Decompose a face into 4 dimensions of features:

1. eye height
2. eye separation
3. nose length
4. mouth height

$X_{i\,k}$

i: the index of face

k: the index of the feature

Response probabilities

$$P(R_i = A|i) = \frac{\left(\sum_{j \in A} s_{ij} \right)}{\left(\sum_{j \in A} s_{ij} \right) + \left(\sum_{j \in B} s_{ij} \right)}$$

```
# likelihood from the Generalized Context Model
source("GCMpred.R")

N <- 2*80    # there were 2 responses per face from 80 subjects, in total 160 responses
N_A <- 155 # the number of A response. N_B is implicitly N - N_A

c <- 4          # a parameter for calculating similarity
w <- c(0.19, 0.12, 0.25, 0.45)   # parameters for the weights

# read the 4d features of 34 face stimuli
stim <- as.matrix(read.table("faceStim.csv", sep=","))
# two categories (a & b) of examplars which have been stored in your memory
exemplars <- list(a=stim[1:5,], b= stim[6:10,])

preds <- GCMpred(stim[1,], exemplars, c, w) # the probability of A response
likelihood <- dbinom(N_A, size = N, prob = preds[1])
```

GCMbinom.R

```
# Prediction of the Generalized Context Model
GCMpred <- function(probe, exemplars, c, w){

  dist <- list()

# calculating the distance between probe and each exemplars
  for (ex in exemplars){
    dist[ [length(dist)+1] ] <- apply(as.array(ex), 1, function(x) sqrt(sum(w*(x-probe)^2)))
  }

# calculating the similarity between probe and category A, between probe and category B
  sumsim <- lapply( dist, function(a) sum(exp(-c*a)) )

  r_prob <- unlist(sumsim)/sum(unlist(sumsim))

}
```

$$d_{ij} = \left(\sum_{k=1}^{K} |x_{ik} - x_{jk}|^2 \right)^{\frac{1}{2}}$$

$$s_{ij} = exp(-c \cdot d_{ij})$$

$$P(R_i = A|i) = \frac{\left(\sum_{j \in A} s_{ij} \right)}{\left(\sum_{j \in A} s_{ij} \right) + \left(\sum_{j \in B} s_{ij} \right)}$$

GCMpred.R

The binomial distribution

$$f(k|p_{heads}, N) = \binom{N}{k} {p_{heads}}^{k} (1 - p_{heads})^{N-k}$$

P(data| p_{heads}) =

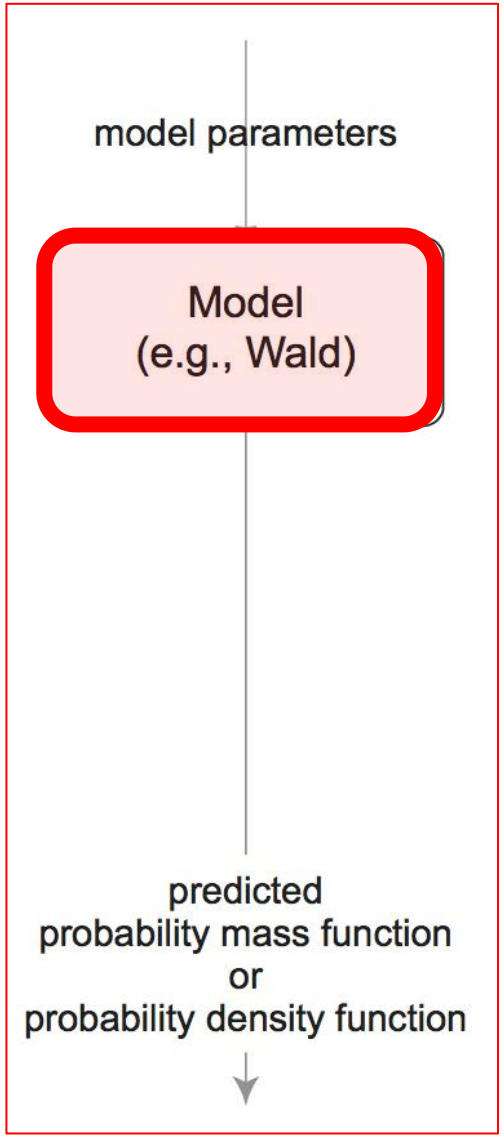
$$\binom{6}{5} {p_{heads}}^{5} (1 - p_{heads})^{1}$$

P(p_{heads}|data) = ?

P($coin$ is fair|data) = ?

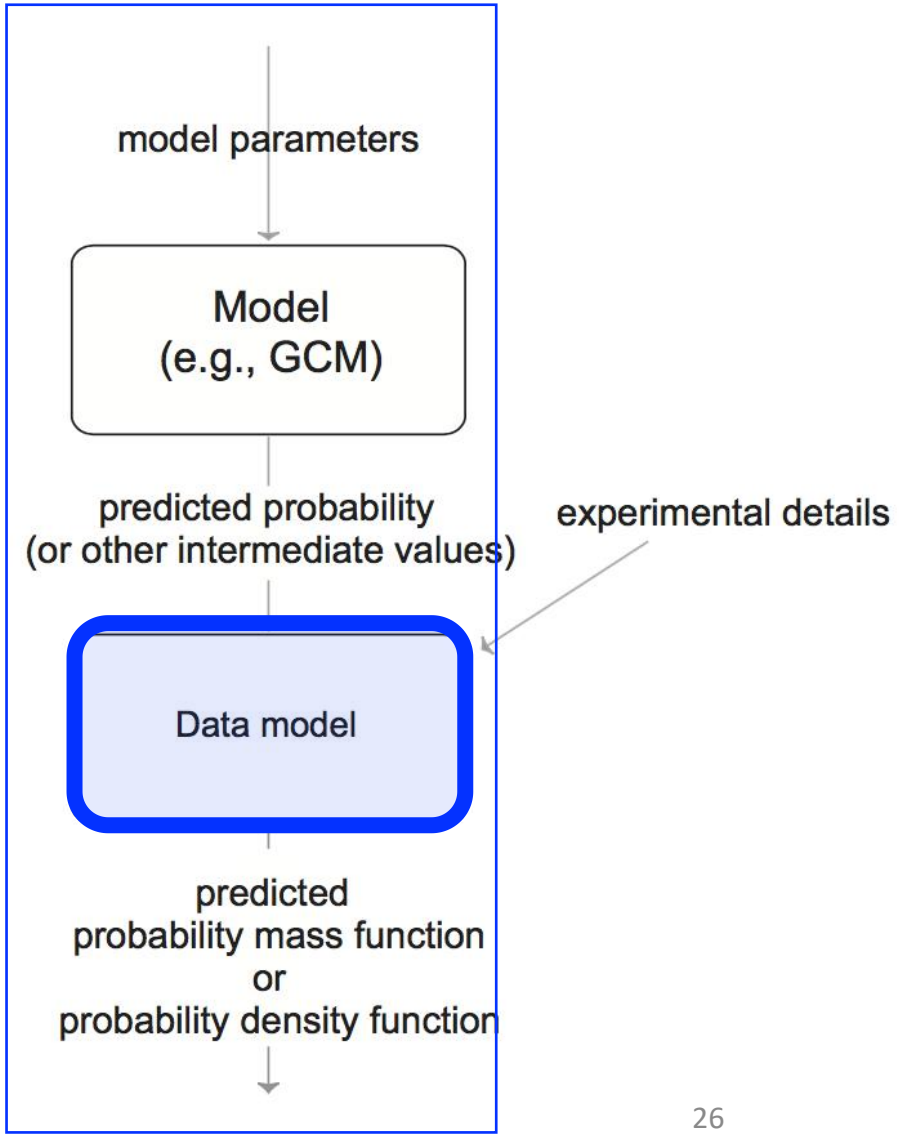
Lecture5_coin_likelihood.R

Two types of probability functions



Directly relate the model to the data

GCM with a binomial data model

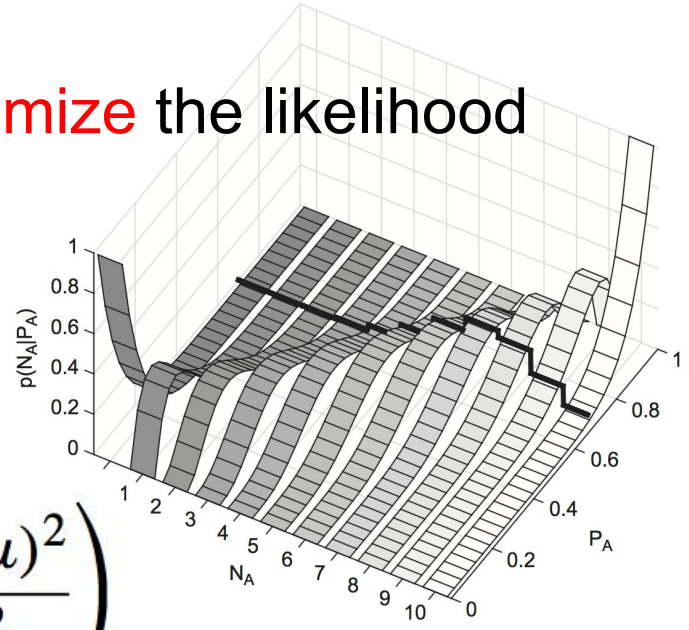


Finding the Maximum Likelihood

Analytical solution: derivative = 0
Grid search: to plot likelihood surfaces, and identify that combination of parameters that gives the highest point on the surface (e.g., Eliason 1993).

Simplex algorithm: to search *the parameter space* for maximize the likelihood function

Instead of likelihood, we usually calculate the log likelihood.



$$f(\mathbf{y}|\boldsymbol{\theta}) = \prod^{k} f(y_k|\boldsymbol{\theta}) \qquad p(y|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$$

$$\ln L(\boldsymbol{\theta}|\mathbf{y}) = \sum_{k=1}^{K} \ln L(\boldsymbol{\theta}|y_k) \qquad \ln L(\mu,\sigma|y) = \ln(1) - \ln(\sqrt{2\pi\sigma^2}) - \frac{(y-\mu)^2}{2\sigma^2}$$

Parameter Estimation with bounds
nmkb function in R

Nelder-Mead optimization algorithm for derivative-free optimization.
It allows *bounds* to be placed on parameters.

Input:

1.Starting values of the model parameters,
2.fn = A *function* to be minimized # such as minimize a negative log likelihood
3. lower = lower bounds
4. upper = upper bounds

Output:

$par: the best-fitting parameter estimates (a list structure)
$value: value of the function under the best fit

fitGCMnoisy.R

Lecture 5 – Maximum Likelihood Estimation

- Motivations for MLE
- Basics of Probabilities: properties of probability; probability functions
- What is a likelihood
- Defining a Probability Distribution
  - Specified by the NeuroPsychological Model
  - Specified by by Data Models (binomial distribution)
- Finding the Maximum Likelihood

Recommended materials

Textbook

- Computational Modeling of Cognition and Behavior, Chapter 4

Must read.

Online courses

- MIT RES.6-012 概率导论 (Introduction to Probability) (Spring 2018), by Prof. John Tsitsiklis

https://www.bilibili.com/video/av74610113?p=1

- MIT 6.041 Probabilistic Systems Analysis and Applied Probability, by Prof. John Tsitsiklis

https://www.bilibili.com/video/av64033499?p=1 (中英字幕)