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 10 – Markov Chain Monte Carlo Methods

**Quanying Liu （刘泉影）**

SUSTech, BME department

Email: liuqy@sustech.edu.cn

# Recap Lecture 9 – Bayesian Parameter Estimation

- What is Bayesian Inference?
  - Motivations
  - Bayes Theorem: prior, likelihood, evidence, posterior
- Analytic Methods for Obtaining Posteriors
  - The likelihood function
  - The Prior Distribution → conjugate prior
  - The Posterior Distribution
  - An example: Estimating the bias of a coin
- Determining the Prior Distributions of Parameters
  - Non-Informative Priors → Jeffreys prior
  - Reference Priors

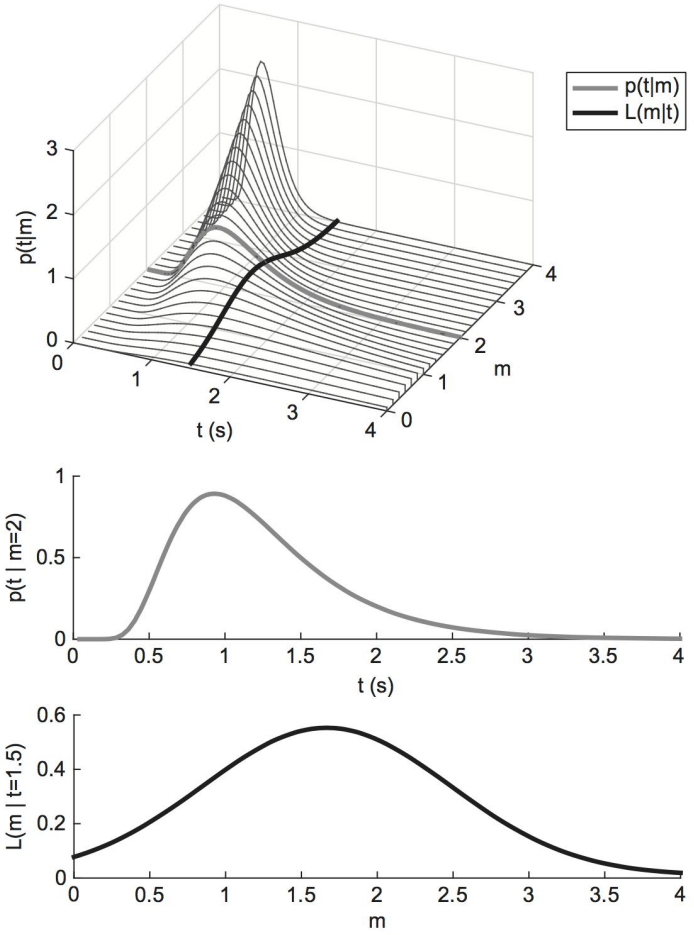
# Motivations for Bayesian Inference

- We have learned Maximum Likelihood Estimation (**MLE**) in Lecture 5.
- Recall **MLE**: $L(\boldsymbol{\theta}|y)$ ← Probability distribution $f(y|\boldsymbol{\theta}, M)$

- It permits to estimate parameter value by ***relative*** comparisons between different parameter values. But it is not suited for estimating ***absolute*** probabilities.
- It did not combine our prior knowledge of the parameters.

**Motivations** for Bayesian Parameter Estimation:

- We want to know is the likely range of the "true" parameter value that we can infer from our measurements (information about the probability distribution of the parameters)
- We want to incorporate our prior knowledge of the parameters into the consideration.



# Bayes Theorem

Prior – what we know about $\boldsymbol{\theta}$ **BEFORE** collecting data y

Likelihood – propensity for observing a certain value of y given a certain value of $\boldsymbol{\theta}$

$$p(\boldsymbol{\theta}|y) = \frac{p(\boldsymbol{\theta})p(y|\boldsymbol{\theta})}{p(y)} = \frac{p(\boldsymbol{\theta})p(y|\boldsymbol{\theta})}{\sum_{\boldsymbol{\theta}} p(\boldsymbol{\theta})p(y|\boldsymbol{\theta})}$$

Posterior – what we know about y **AFTER** seeing y

Evidence – a **constant** to ensure that the left hand side is a valid distribution

$$p(\boldsymbol{\theta}|y) \propto p(\boldsymbol{\theta})p(y|\boldsymbol{\theta})$$

# The likelihood function

- Bernoulli $f(x = k|\theta) = \theta^k(1-\theta)^{1-k}$
- Categorical $f(x = e_k|\boldsymbol{\theta}) = \theta_k$
- Binomial $f(x = k|n, \theta) = \binom{n}{k}\theta^k(1-\theta)^{n-k}$
- Poisson $f(x = k|\theta) = e^{-\theta}\frac{\theta^k}{k!}$

- Gaussian $f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ , $\boldsymbol{\theta} = [\mu, \sigma^2]$

# The application: Estimating the bias of a coin

- Beta prior $\text{Beta}(\theta|a,b) \propto \theta^{a-1}(1-\theta)^{b-1}$
- Binomial likelihood $\text{Bin}(y=k|n,\theta) = \binom{n}{k}\theta^k(1-\theta)^{n-k}$
- Posterior distribution $p(\theta|y=k) = \text{Beta}(\theta|a+k, b+n-k)$

- We do an experiment, and toss a coin **10** times
- We observe **6** heads, **4** tails.
- What is the posterior distribution of $\theta$?
  1) Assuming the prior distribution of $\theta$ as $\text{Beta}(\theta|1,1)$, the posterior is $\text{Beta}(\theta|7,5)$
  2) Assuming the prior distribution of $\theta$ as $\text{Beta}(\theta|12,12)$, the posterior is $\text{Beta}(\theta|18,16)$
  3) Assuming the prior distribution of $\theta$ as $\text{Beta}(\theta|1,4)$
  4) Assuming the prior distribution of $\theta$ as $\text{Beta}(\theta|4,1)$
  5) Assuming the prior distribution of $\theta$ as $\text{Beta}(\theta|100,100)$

# Lecture 10 – Markov Chain Monte Carlo (MCMC)

- What is MCMC?
  - Motivations
  - The Metropolis-Hastings Algorithm for MCMC
  - Estimating single parameter
  - Estimating multiple parameters
- Two major problems for MCMC sampling
  - Convergence of MCMC Chains
  - Autocorrelation in MCMC Chains
- Approximate Bayesian Computation (ABC): a likelihood-free method
  - Likelihoods that cannot be computed
  - From simulations to estimates of the posterior
  - An Example: applying ABC in a single-detection model

# Motivations for MCMC methods

We have introduced the beta-binomial model, where the posterior distribution has an analytic solution.

For some complex models, the posterior distribution is not analytically tractable.

The **concept** for MCMC is that to sample from a distribution.
Given a sufficiently **large** sample, we can begin to understand many attributes of that distribution, even if we do not have a specific formula for it.

An unknown distribution is approximated by a large sample:

```
rbeta(10000, 0.5,  0.5)
rnorm(10000,   0,    1)
…
```

# MCMC methods

Recall the Baye's rule: $p(\boldsymbol{\theta}|y) \propto p(\boldsymbol{\theta})p(y|\boldsymbol{\theta})$

So long as we are able to compute $p(\boldsymbol{\theta})p(y|\boldsymbol{\theta})$, we can sample from the posterior distribution using a technique known as Markov Chain Monte Carlo (MCMC).

A **Markov Chain** is formed by a random process that undergoes transitions between states, where the state at time t+1 depends only on the state at time t.

It is ahistorical or "**memory-less**", because its current state is entirely independent of what happened more than one step in the past.

In MCMC, likewise, we probabilistically **transition** from one state to another, and although successive pairs of states are therefore correlated with each other, states that are more than one step apart are increasingly closer to independence as their separation increases.

# 3 approaches to Bayesian Parameter Estimation

| Knowledge required | Analytic Methods (Chapter 6) | Monte Carlo Methods (Section 7.1) | Approximate Bayesian Computation (Section 7.3) |
|---|---|---|---|
| Prior distribution | Assumed | Assumed | Assumed |
| Likelihood | Computed and known | Computed and known | Cannot be computed but results can be simulated |
| Posterior distribution | Derived analytically<br>• $p(\theta \mid y)$ can be fully evaluated and integrated | Sampled by MCMC<br>• $p(\theta \mid y)$ can be evaluated up to a proportionality constant | Sampled by comparing data to candidate simulation results<br>• neither $p(\theta \mid y)$ nor $p(y \mid \theta)$ need to be computable |

# Metropolis-Hastings Algorithm for MCMC

Metropolis-Hastings Algorithm is one of **the top 10 most influential algorithms** of the 20th century (Beichel and Sullivan, 2000).

The algorithm is readily summarized:

- We take a guess, add some random noise, and if that improves our guess we accept the answer and move on.
- If the guess does not get better with noise, we probably reject the answer and proceed with our initial guess.
- When we have done this many times, our accepted guesses form the desired sample from the posterior.

# Metropolis-Hastings Algorithm for MCMC

1. Create a plausible starting value for the parameter(s). This becomes our current sample. Its posterior distribution is the target distribution for MCMC.
2. Draw random noise from a proposal distribution, and add it to the current sample to obtain a proposal. The proposal distribution **must** be *zero-centered* and *symmetrical* to permit fluctuations in either direction.
3. Compare the value of the target distribution at the proposal to the value of the target distribution at the current sample.
   a. If the proposal is associated with a **greater** value than the current sample, accept it.
   b. If the proposal is associated with a **smaller** value, accept it with a probability equal to the ratio of the two values, otherwise reject it.
4. If the proposal has been **accepted**, current <- proposal. If the proposal has been **rejected**, the current sample is reused as the next sample in the chain.
5. Return to Step 2, and continue the process until enough samples have been obtained.

# Metropolis-Hastings Algorithm for MCMC

1. Initialise $x^{(0)}$.
2. For $i = 0$ to $N - 1$
   - Sample $u \sim \mathcal{U}_{[0,1]}$.
   - Sample $x^\star \sim q(x^\star | x^{(i)})$.
   - If $u < \mathcal{A}(x^{(i)}, x^\star) = \min\left\{1, \frac{p(x^\star)q(x^{(i)}|x^\star)}{p(x^{(i)})q(x^\star|x^{(i)})}\right\}$

     $$x^{(i+1)} = x^\star$$

     else

     $$x^{(i+1)} = x^{(i)}$$

C Andrieu et al (2003) An Introduction to MCMC for Machine Learning

# A simple example with a uniform prior

Suppose we administer an intelligence test to a SUSTech student who is one of a group of children considered to be particularly gifted.

This student scores $y = 144$ on a test that is known to be normally distributed with $\sigma = 15$ for the population at large.

What is the likely value of $\mu$ for the population of gifted children, assuming a uniform prior distribution for $\mu$?

$$p(\boldsymbol{\theta}|y) \propto p(\boldsymbol{\theta})p(y|\boldsymbol{\theta})$$

SEE CODE for the implementation:

Lecture10_1_Metropolis_Hastings.R

```
#perform MCMC
burnin<-200

chain <- rep(0,5000)
obs <- 144      # observed data
propsd <- 2     # tuning parameter

chain[1] <- 150  #starting value
for (i in 2:length(chain)) {
    current <- chain[i-1]
    proposal <- current + rnorm(1,0,propsd)
    if (dnorm(obs,proposal,15) > dnorm(obs,current,15)) {
       chain[i] <- proposal  #accept proposal
    } else {
       chain[i] <- ifelse(runif(1) < dnorm(obs,proposal,15)/dnorm(obs,current,15),
                          proposal, #accept proposal
                          current)   #reject proposal
    }
}
```

# A simple example with a Gaussian prior

Suppose we administer an intelligence test to a SUSTech student who is one of a group of children considered to be particularly gifted.

This student scores $y = 144$ on a test that is known to be normally distributed with $\sigma = 15$ for the population at large.

What is the likely value of $\mu$ for the population of gifted children, assuming a Gaussian prior distribution for $\mu$ with mean=103, sd=20?

$$p(\boldsymbol{\theta}|y) \propto p(\boldsymbol{\theta})p(y|\boldsymbol{\theta})$$

SEE CODE for the implementation:

Lecture10_1_Metropolis_Hastings.R

# Estimating multiple parameters

MCMC methods can be generalized to **multi-parameter** models.

Take an example from a visual working memory study.

Two theories about the limits on the storage capacity of working memory

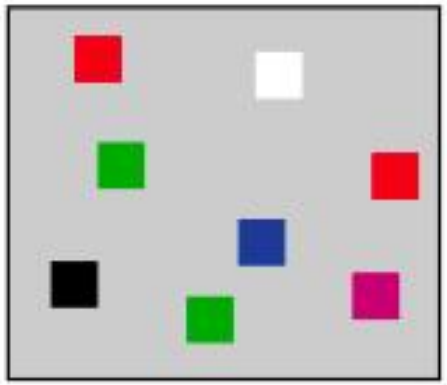
- **Theory 1**: working memory stores a limited set of **discrete**, **fixed-resolution** representations
- **Theory 2**: flexibly to provide either a **small** number of **high-resolution** representations or a **large** number of **low-resolution** representations

Vote: 1 or 2

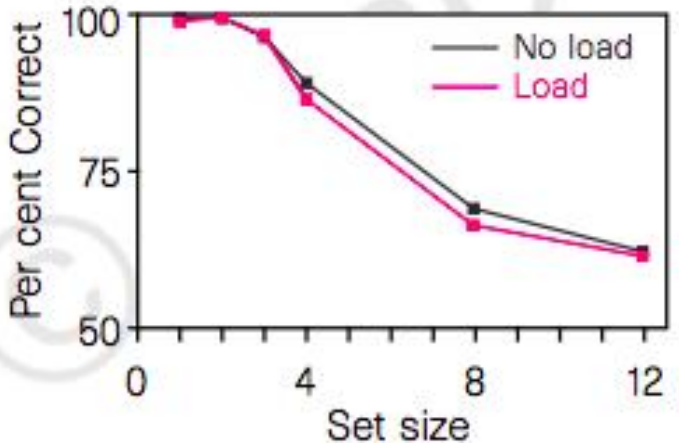
Luck & Vogel (1997) Nature

# Estimating multiple parameters

**Theory 1**: working memory stores a limited set of discrete, fixed-resolution representations



100ms



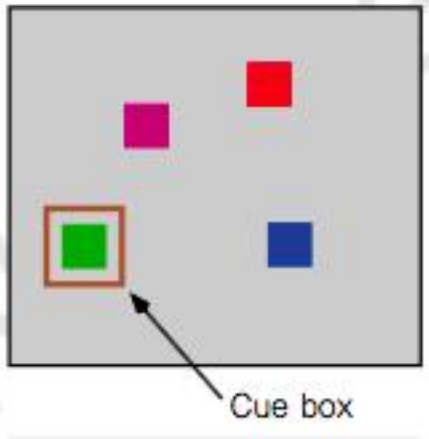

- The first set of experiments is **to examine working memory capacity for simple colours**.
- The sample array consisted of 1–12 coloured squares and was presented for 100 ms.
- This was followed by a 900-ms blank delay interval, and then a 2,000-ms presentation of the test array, which was either identical to the sample array or differed in the colour of one of the squares.

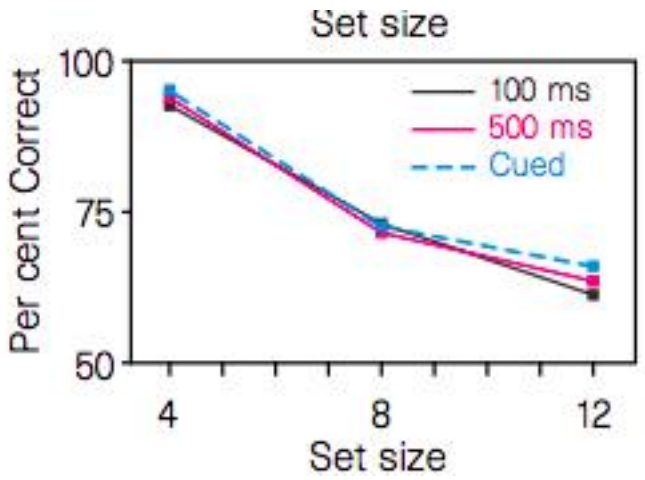
Findings:
Performance was nearly perfect for arrays of 1–3 items and then declined systematically as the set size increased from 4 to 12 items.
According to the method for estimating memory capacity described by Pashler [9], these data indicate that the observers were able to retain the colours of roughly **four items** in working memory, which is similar to previous estimates for alphanumeric characters [21].

Luck & Vogel (1997) Nature

# Estimating multiple parameters

**Theory 1**: working memory stores a limited set of discrete, fixed-resolution representations



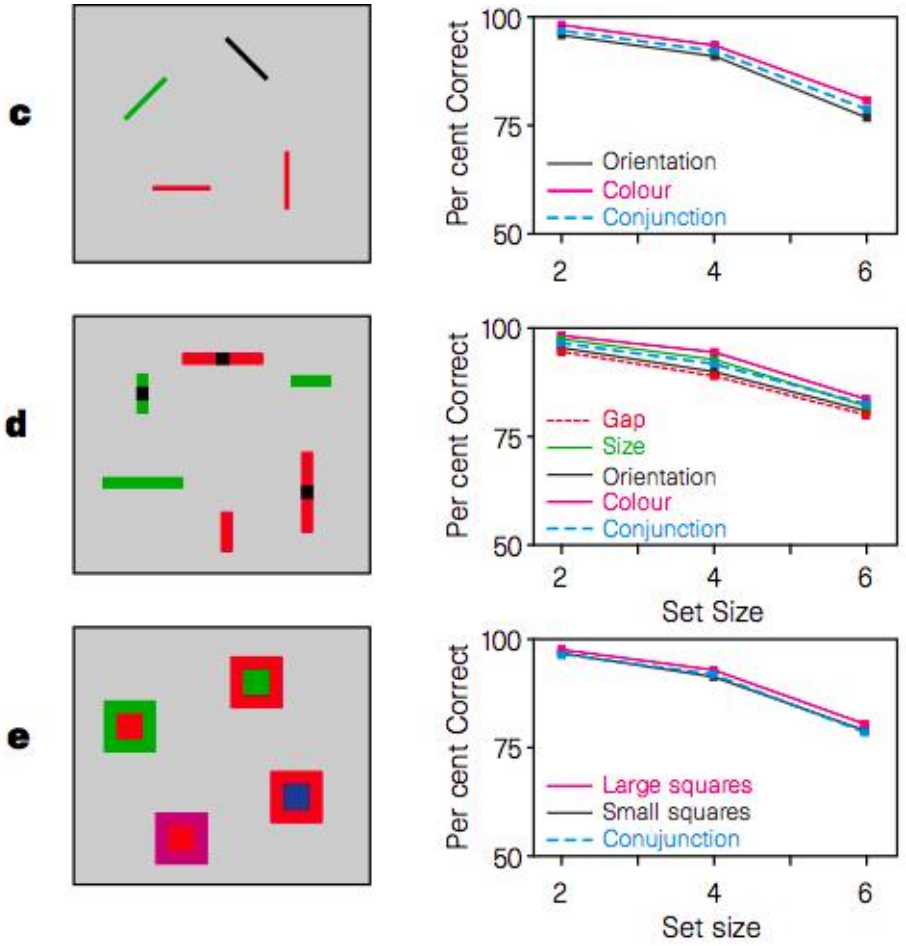

- It was also necessary to demonstrate that the relatively small memory capacity observed in this experiment was **not a result of limitations in processes** other than working-memory storage.
- To rule out limitations in perceiving the stimuli and encoding them in working memory, we varied the duration of the sample stimulus, comparing the original 100-ms duration with a **500-ms** duration.
- This allowed more time for perceiving the stimuli and encoding them in memory, which should have led to improved performance if these were limiting factors.

Findings:
performance was **not** significantly influenced by variations in sample duration (Fig. 1b), indicating that the errors at set sizes of **4–12 reflected limitations in storage capacity** rather than limitations in perceiving or encoding the stimuli.

Luck & Vogel (1997) Nature

# Estimating multiple parameters

**Theory 1**: working memory stores a limited set of discrete, fixed-resolution representations




- To determine **whether capacity is different for different feature dimensions**, memory for orientation was compared with memory for colour using 4, 8 or 12 bars that varied both in colour and in orientation.
- In one condition, **only colour** could vary between the sample array and the test array, and the observers were instructed to look for a colour change. In a second condition, **only orientation** could vary. In the third and critical condition, **either colour or orientation** could vary, and the observers were required to remember both features of each object.

- The surprisingly good performance for conjunctions could be explained by the use of separate, **independent memory systems for each feature type** rather than the storage of integrated object representations.

Luck & Vogel (1997) Nature

# Estimating multiple parameters

**Theory 1**: working memory stores a limited set of discrete, fixed-resolution representations

Conclusion:

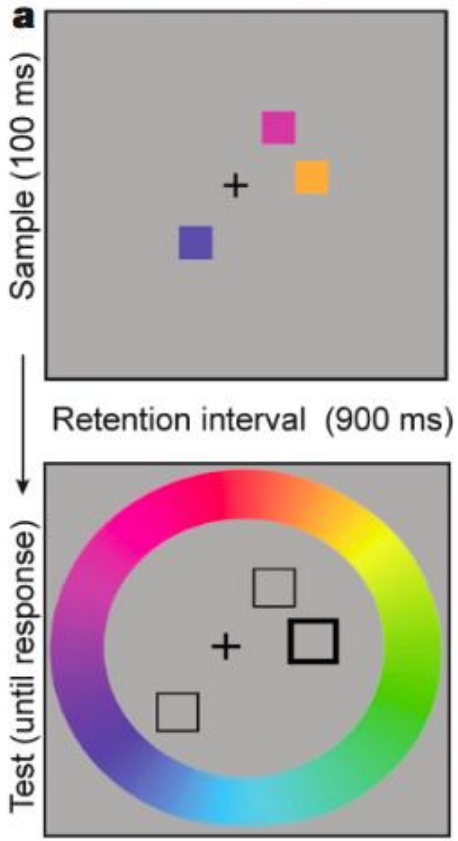
These results indicate that **integrated object percepts** are stored in visual working memory, leading to a large capacity for retaining individual features as long as the features are confined to a small number of objects.

Although there may be **limits** on the number of features that can be linked together in a single object representation, our results indicate that at least four features can be joined in this manner with no cost in terms of storage capacity.

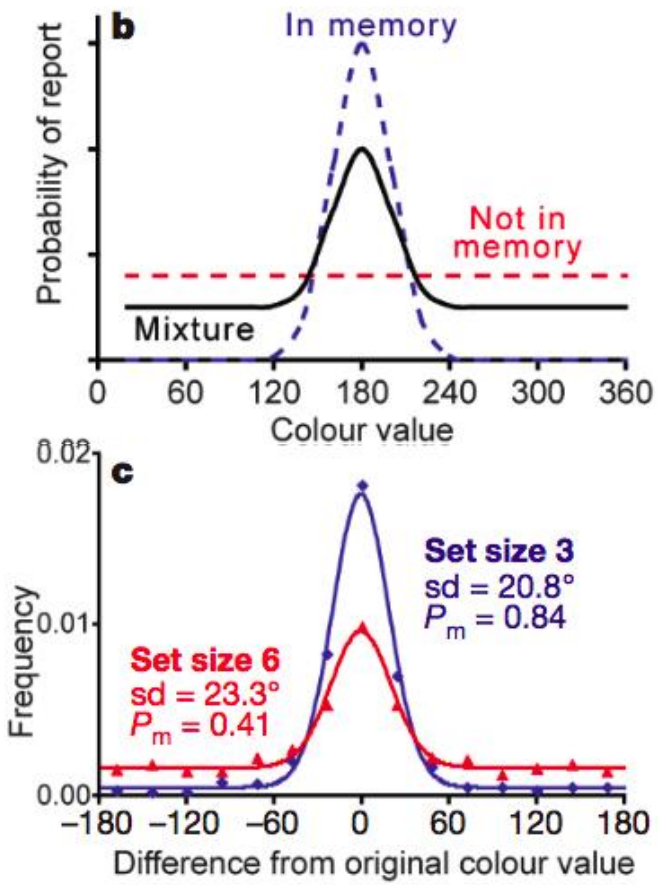
Luck & Vogel (1997) Nature

# Estimating multiple parameters

**Theory 2**: flexibly to provide either a small number of high-resolution representations or a large number of low-resolution representations




Task




**Mixture Models**

If the probed item has been stored in working memory, the recalled value will tend to be *near* the original color with a limited number of 'slots';

If the probed item has not been stored, then the observer will have no information about the color, and the responses should be *random*.

Parameter (**Pm**): the probability that the probed item was present in memory;

Parameter (**sd**): the precision of the representation when the cued item was present in memory.

Zhang & Luck (2008) Nature

# Estimating multiple parameters

**Theory 2**: flexibly to provide either a small number of high-resolution representations or a large number of low-resolution representations

SEE CODE: Lecture10_3_mixModel.R

Parameter (**g**): the probability that the probed item was not present in memory, which means the weighting of the probability densities for a uniform distribution U(-180, 180)

Parameter (**sdv**): the precision of the representation when the cued item was present in memory. It describes the standard deviation of *a "circular" Gaussian distribution* with *mean zero*, called von Mises distribution(冯·米塞斯分布), to describe responses that are **not guesses** but reflect **memory** for the item.

$$f(x|\mu,\kappa) = \frac{e^{\kappa\cos(x-\mu)}}{2\pi I_0(\kappa)}$$

where $I_0(x)$ is the modified Bessel function of order 0.

**Likelihood:** g*dunif(data,-180,180)) + (1-g)* dvonmises(data4vm, mkcirc(0), sd2k(sdv))

Zhang & Luck (2008) Nature

# Two major problems for MCMC sampling

## 1. **Convergence** of MCMC chains

- Convergence means that a chain tours the entire distribution and the exploration of the distribution is no longer dependent on starting values.
- The cases that convergence might fail:
  a. multimodal posterior: two peaks and a shallow valley in between
  b. the detection of outliers in a regression situation
- No guaranteed solution to the convergence problem, although there are **several safeguards** that we can take (i.e. the use of multiple chains which originates from different starting values).

## 2. **Autocorrelation** in MCMC chains

- Autocorrelation problem: successive samples in an MCMC chain must be correlated
- It is **only** when one considers samples that are spaced further apart along the chain that they become essentially independent.
- One solution to the autocorrelation problem is a process known as "thinning". considering every *k*th sample ($2 < k < 40$) → not efficient

# Approximate Bayesian Computation (ABC)

- **ABC** allows us to apply Bayesian techniques to parameter estimation, when the *likelihood*, $p(y|\boldsymbol{\theta})$, cannot be computed.

- **Likelihood in the mixture model:**

g*dunif(data,-180,180)) + (1-g)* dvonmises(data4vm, mkcirc(0), sd2k(sdv))

- Sometimes, we might have no clear formulas for the likelihood function.

- **ABC** replaces computation of the likelihood function with **a simulation of the model** in question. We replace $p(y|\boldsymbol{\theta})$ with its *simulated approximation*, as we have done in MCMC.
- We describe the simulation by the notation $\eta(\theta)$

# From Simulations to Estimates of the Posterior

## Rejection Algorithm

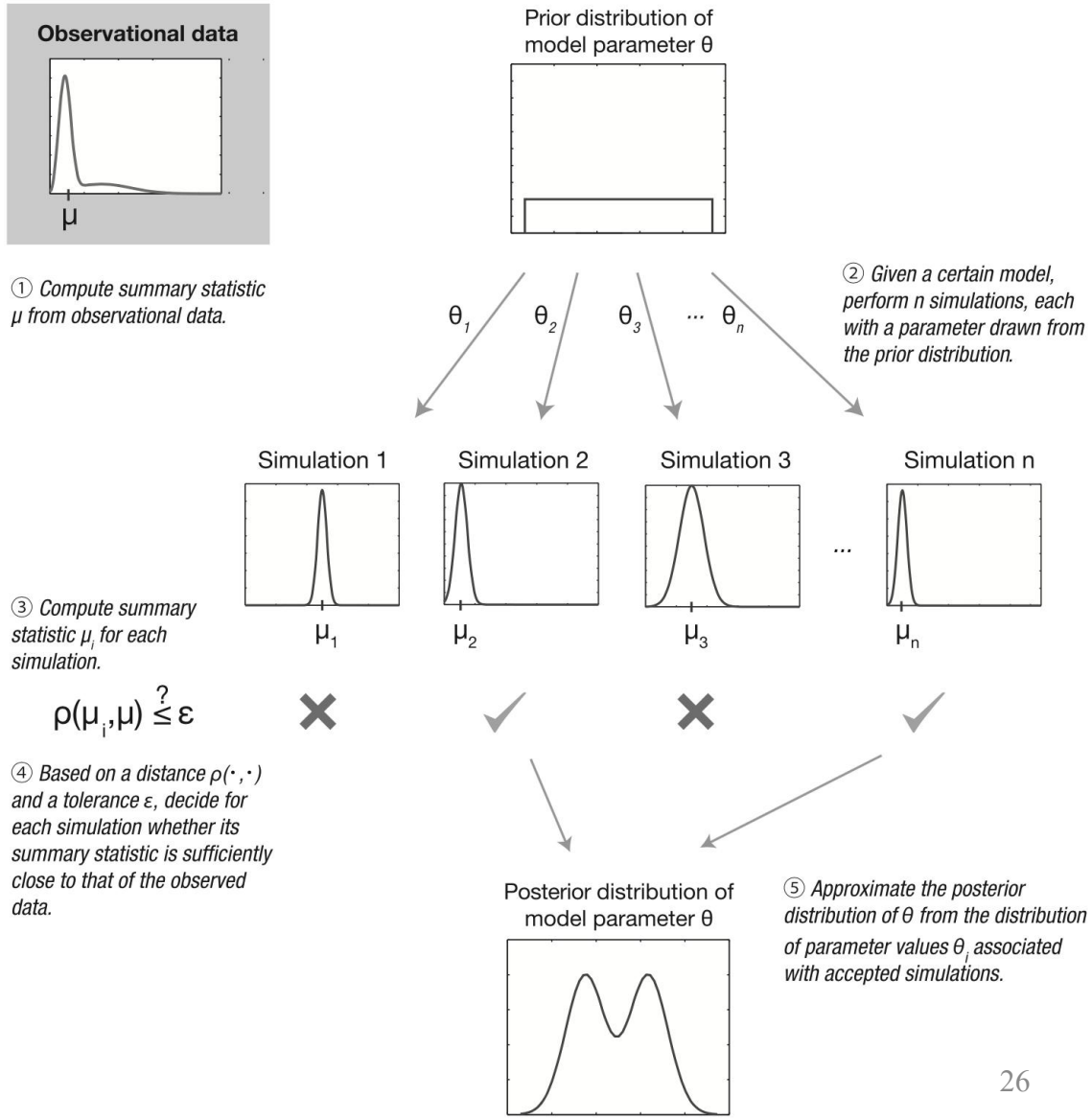
Set $i = 1$

Repeat until $i = N$

1. Sample $x^{(i)} \sim q(x)$ and $u \sim \mathcal{U}_{(0,1)}$.
2. If $u < \frac{p\left(x^{(i)}\right)}{Mq\left(x^{(i)}\right)}$ then accept $x^{(i)}$ and increment the counter $i$ by
   1. Otherwise, reject.

C Andrieu et al (2003)

- distance function
- tolerance



# Signal Detection Theory

**Table 11.1** Basic signal detection theory data and terminology.

| | Signal trial | Noise trial |
|---|---|---|
| Yes response | Hit | False alarm |
| No response | Miss | Correct rejection |

**Two** assumptions for signal-detection theory:

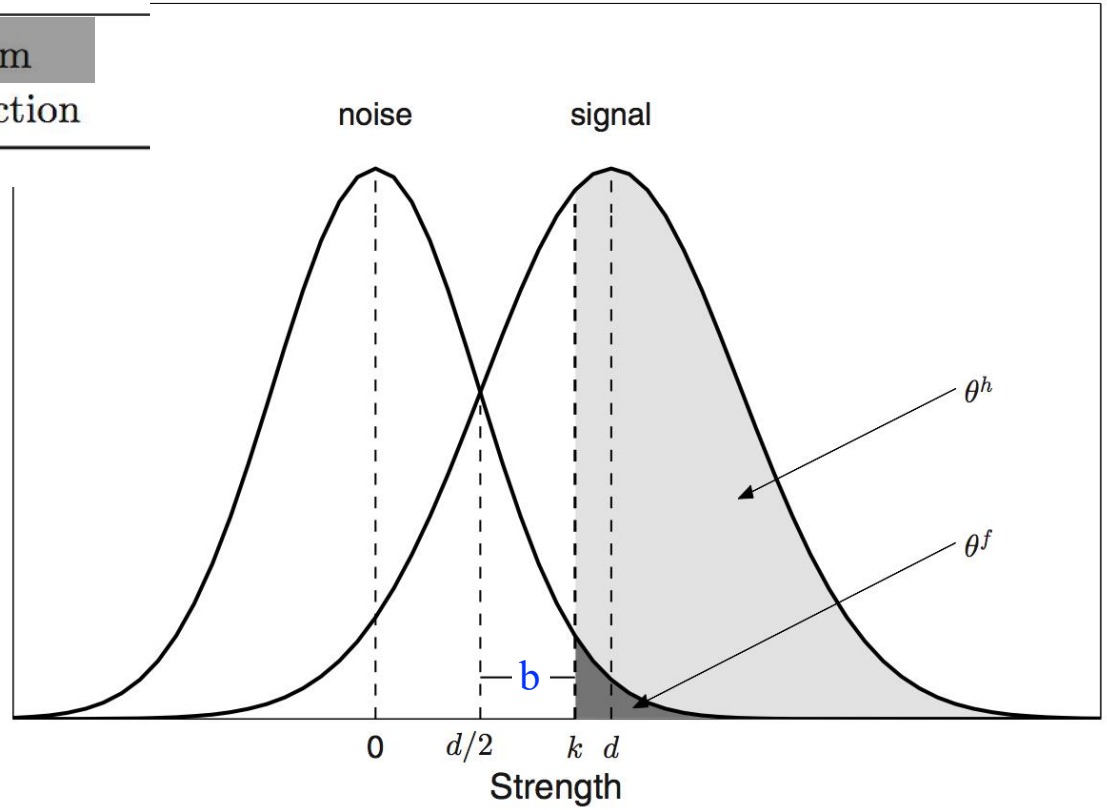
1) **Signal** and **noise** trials can be represented as values along a uni-dimensional "strength" construct.

2) Both types of trials produce strengths that vary according to a Gaussian distribution along this dimension.

d: discriminability of the signal trials from the noise trials

k: criterion for response

b: bias (b = k-d/2)




Equal-variance Gaussian signal detection theory framework.

d/2: the unbiased criterion

# Applying ABC in a single-detection model

## A recognition memory task

- Study session: Participants first study a list of words and are then presented with a long sequence of test items.
- Test session: Each test item is *either* a word from the study list (old) *or* an item that has not been seen before (new). Old and new items usually appear with equal probability in the test sequence. The participant *responds* to each test item by indicating “old” or “new”.

| | Test Item: Old | Test Item: New |
|---|---|---|
| Response: “Old” | 60% | 11% |
| Response: “New” | 40% | 89% |

**Table 11.1** Basic signal detection theory data and terminology.

| | Signal trial | Noise trial |
|---|---|---|
| Yes response | Hit | False alarm |
| No response | Miss | Correct rejection |

# SEE CODE: Lecture9_5_abcsdt.R

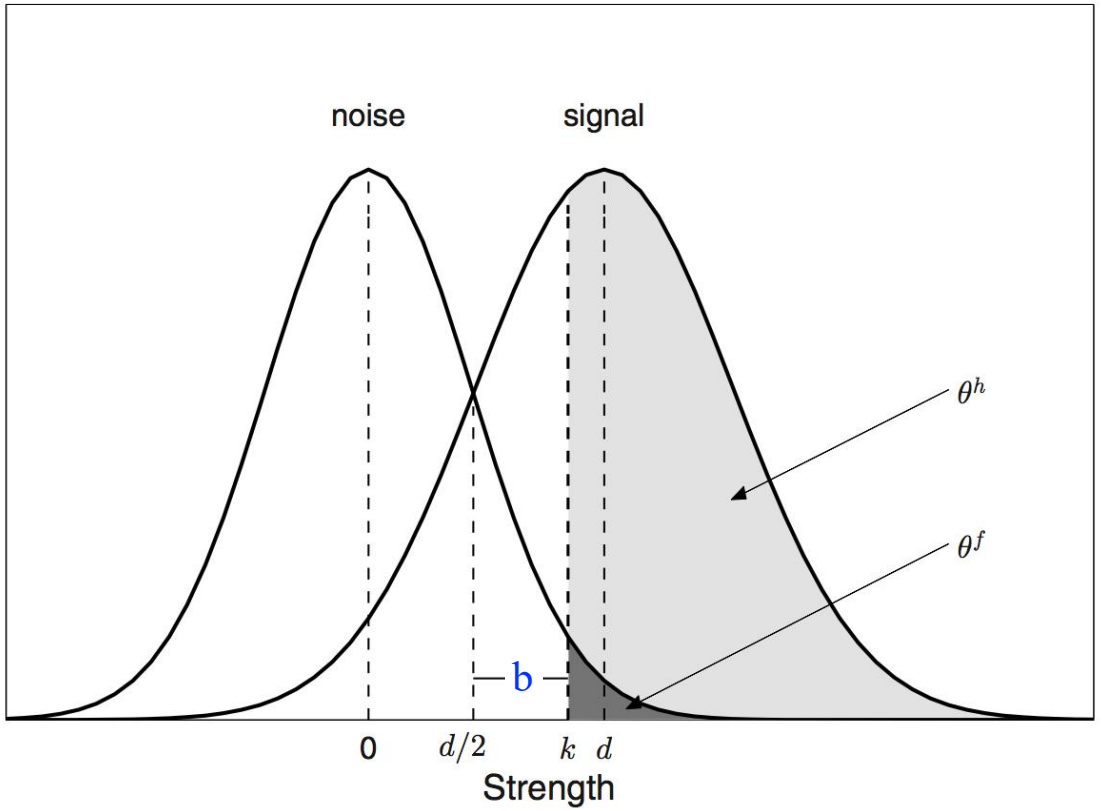
```
y    <- c(60,11)  #define target data
dmu <- 1          #define hyperparameters
bmu <- 0
dsigma <- bsigma <- 1

ntrials <- 100
epsilon <- 1
posterior <- matrix(0,1000,2)
for (s in c(1:1000)) {  #commence ABC
  while(TRUE) {
    dprop <- rnorm(1,dmu,dsigma)
    bprop <- rnorm(1,bmu,bsigma)
    X<-simsdt(dprop,bprop,ntrials) #simulate proposal
    if (sqrt(sum((y-X)^2)) <= epsilon) {break}
  }
  posterior[s,]<-c(dprop,bprop) #keep good simulation
  print(s)                      #show sign of life
}
```

```r
#simulate sdt given parameters and number of trials
simsdt<- function(d,b,ntrials) {
  old <- rnorm(ntrials/2,d)
  hits <-sum(old >(d/2+b))/(ntrials/2)*100
  new <- rnorm(ntrials/2,0)
  fas <- sum(new>(d/2+b))/(ntrials/2)*100
  return(X<-c(hits,fas))
}
```

Criterion: in the simulated trials, pro > d/2 + b

Return X: Percentage of hits and false alarm

**X is simulated from the SDT model**




Equal-variance Gaussian signal detection theory framework.

# Summary for Bayesian Parameter Estimation

| Knowledge required | Analytic Methods (Chapter 6) | Monte Carlo Methods (Section 7.1) | Approximate Bayesian Computation (Section 7.3) |
|---|---|---|---|
| Prior distribution | Assumed | Assumed | Assumed |
| Likelihood | Computed and known | Computed and known | Cannot be computed but results can be simulated |
| Posterior distribution | Derived analytically<br>• $p(\theta \mid y)$ can be fully evaluated and integrated | Sampled by MCMC<br>• $p(\theta \mid y)$ can be evaluated up to a proportionality constant | Sampled by comparing data to candidate simulation results<br>• neither $p(\theta \mid y)$ nor $p(y \mid \theta)$ need to be computable |

# Lecture 9 – Summary

- What is MCMC?
  - Motivations
  - The Metropolis-Hastings Algorithm for MCMC
  - Estimating single parameter (examples from the intelligence tests)
  - Estimating multiple parameters (an example from the visual working memory)
- Two major problems for MCMC sampling
  - Convergence of MCMC Chains → multiple chains
  - Autocorrelation in MCMC Chains → thinning
- Approximate Bayesian Computation (ABC): a likelihood-free method
  - Likelihoods that cannot be computed
  - From simulations to estimates of the posterior
  - An Example: applying ABC in a single-detection model

# Reading materials

## Textbooks

- Chapter 7 (Markov Chain Monte Carlo Methods)

## Papers

- Zhang and Luck (2008). Discrete fixed-resolution representations in visual working memory, Nature
- C Andrieu, N de Freitas, A Doucet, M Jordan, (2003) An Introduction to MCMC for Machine Learning, Machine Learning

- Extra readings for *von Mises distribution* and *signal-detection model*