Statistical Methods for Data Science & Lab (Part I)

Topic 13

Parametric Inference and the Parametric Bootstrap

Pierpaolo Brutti
Department of Statistics
Sapienza University

Outline

Agenda

- Parametric Models & the Parameter of Interest
- The Method of Moments & its properties
- The Likelihood function and the Maximum Likelihood Estimator
- Computing the Maximum Likelihood Estimator
- Properties of the Maximum Likelihood Estimator
- Asymptotic Normality and Confidence Intervals
- The δ–Method
- The Parametric Bootstrap
- Model Selection: AIC and Cross–Validation

Main References

- Ch.19, Ch.20 and Ch.21 of the **blue** book.
- Ch.09 of the **purple** book.

Generalities

...it's a small small parametric world out there...

Parametric Inference

Introduction

- We turn our attention to parametric models, i.e. models of the form

  $$\mathcal{F} = \left\{ f(x \mid \boldsymbol{\theta}) \,:\, \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^k \right\} \quad \text{(typically } k << n\text{)}$$

  Sometimes the sampling model will also be denoted by $f_{\boldsymbol{\theta}}(\cdot)$ for short.

- The problem of inference then reduces to the problem of estimating the unknown euclidean parameter vector $\boldsymbol{\theta}$.

- Parametric model are *rarely correct*, they may be *useful simplification* of the data generating process at best, which is why nonparametric methods are often preferable.

- In spite of this, whenever the constraints on the data generating process encoded by the parametric model we picked turn out to be reasonable and highly compatible with the experimental evidence, we clearly gain in statistical efficiency by knocking down the variability of our inferential procedures (compared to their nonparametric counterparts).

- Remember that in this setup often but **not** always the parameter $\boldsymbol{\theta}$ has a clear interpretation in terms of population summaries of interest: it may just be a "fictitious" quantity we have to tune to fit the data.

Parametric Inference

Parameter of interest and nuisance parameter

- Often, we are only interested in some function $\tau = h(\boldsymbol{\theta})$ of the parameter.
- For example, if $X \sim \mathrm{N}(\mu, \sigma^2)$ then the parameter is $\boldsymbol{\theta} = [\mu, \sigma^2]^{\mathtt{T}} \in \mathbb{R} \times \mathbb{R}^+$.
- If our goal is to estimate μ then $\mu = h(\boldsymbol{\theta})$ is called the parameter of interest and σ^2 is called a nuisance parameter.
- The parameter of interest might be a very complicated function of $\boldsymbol{\theta}$.

Example

- Let $\{X_1, \dots, X_n\}$ IID $\mathrm{N}(\mu, \sigma^2)$, so $\boldsymbol{\theta} = [\mu, \sigma^2]^{\mathtt{T}} \in \mathbb{R} \times \mathbb{R}^+ = \Theta$.
- Suppose that X_i is the outcome of a blood test and we are interested in $\tau = \{\text{fraction of the population whose test is } > 1\}$.
- Finally, let $Z \sim \mathrm{N}(0, 1)$, then

$$\tau = \mathbb{P}(X > 1) = 1 - \mathbb{P}(X \leqslant 1) = 1 - \mathbb{P}\left(\frac{X-\mu}{\sigma} \leqslant \frac{1-\mu}{\sigma}\right) = 1 - \Phi\left(\frac{1-\mu}{\sigma}\right)$$

Hence the parameter of interest $\tau = 1 - \Phi\left(\frac{1-\mu}{\sigma}\right) = h(\mu, \sigma^2)$.

Method of Moments

...opening the estimator “factory”...

Method of Moments Estimators

Definition

The first method for generating parametric estimators is the method of moments. These estimators are suboptimal but often (not always!) easy to compute.

The Method of Moments

- Let $\boldsymbol{X}_n = \{X_1, \ldots, X_n\}$ be IID with CDF $F(x \mid \boldsymbol{\theta})$.
- Suppose that the parameter $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)$ has k components.
- For $j \in \{1, \ldots, k\}$, define the j^{th} (population) moment as

$$\mu_j \equiv \mu_j(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}}(X^j) = \int x^j \, \mathrm{d}F(x \mid \boldsymbol{\theta})$$

- For $j \in \{1, \ldots, k\}$, define the j^{th} empirical moment as the plug–in

$$\widehat{m}_j \equiv \widehat{m}_j(\boldsymbol{X}_n) = \frac{1}{n} \sum_{i=1}^{n} X_i^j .$$

The method of moments estimator $\widehat{\boldsymbol{\theta}}_n$ is defined to be the value of $\boldsymbol{\theta}$ that solve the following system of k random equations in k unknowns:

$$\widehat{\boldsymbol{\theta}}_n = \underset{\boldsymbol{\theta} \in \Theta}{\mathrm{solve}} \, \mathcal{S}_n(\boldsymbol{\theta} \mid \boldsymbol{X}_n), \quad \text{where} \quad \mathcal{S}_n(\boldsymbol{\theta} \mid \boldsymbol{X}_n) = \begin{cases} \mu_1(\boldsymbol{\theta}) & = \widehat{m}_1(\boldsymbol{X}_n) \\ & \vdots \\ \mu_k(\boldsymbol{\theta}) & = \widehat{m}_k(\boldsymbol{X}_n) \end{cases}$$

Thus the parameter vector is chosen such that the sample moments (on the right side) match the theoretical moments.

Method of Moments Estimators

Examples

Normal model / Both parameters unknown

Let $\boldsymbol{X}_n = \{X_1, \ldots, X_n\}$ be IID from a $\mathrm{N}_1(\mu, \sigma^2)$. Then $\boldsymbol{\theta} = (\mu, \sigma^2)$ and

$$\mu_1 = \mathbb{E}_{\boldsymbol{\theta}}(X) = \mu, \quad \text{and} \quad \mu_2 = \mathbb{E}_{\boldsymbol{\theta}}(X^2) = \mathbb{V}\mathrm{ar}_{\boldsymbol{\theta}}(X) + \big(\mathbb{E}_{\boldsymbol{\theta}}(X)\big)^2 = \sigma^2 + \mu^2.$$

We need to solve the equations

$$\begin{cases} \mu & = \frac{1}{n}\sum_{i=1}^n X_i, \\ \sigma^2 + \mu^2 & = \frac{1}{n}\sum_{i=1}^n X_i^2. \end{cases} \quad \Leftrightarrow \quad \hat{\mu} = \bar{X}_n \quad \text{and} \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^n (X_i - \bar{X}_n)^2.$$

Normal model / Known mean

Let $\boldsymbol{X}_n = \{X_1, \ldots, X_n\}$ be IID from a $\mathrm{N}_1(\mu_0, \sigma^2)$ with μ_0 known.

- There is only one free parameter here, namely $\theta = \sigma^2$, the 1st moment equation $\mu_0 = \bar{X}_n$ carries no information about it since it doesn't depend on σ^2.
- The second equation, instead, brings to the estimator we have already seen

$$\sigma^2 + \mu_0^2 = \frac{1}{n}\sum_{i=1}^n X_i^2 \quad \Leftrightarrow \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^n X_i^2 - \mu_0^2,$$

which can be negative...not that good for a variance!

Method of Moments Estimators

Properties (I)

If $\widehat{\boldsymbol{\theta}}_n$ denote the method of moments estimator, under appropriate conditions on the model, we have

- The estimate $\widehat{\boldsymbol{\theta}}_n$ exists with probability tending to 1.
- The estimate is consistent[1] $\widehat{\boldsymbol{\theta}}_n \xrightarrow{\mathrm{P}} \boldsymbol{\theta}$.
- The estimate is asymptotically Normal:

$$\sqrt{n}\big(\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}\big) \rightsquigarrow \mathrm{N}(0, \Sigma),$$

  where

  - $\Sigma = \boldsymbol{\omega}\, \mathbb{E}_{\boldsymbol{\theta}}\big(\boldsymbol{Y} \cdot \boldsymbol{Y}^{\mathrm{T}}\big)\boldsymbol{\omega}^{\mathrm{T}}$,
  - $\boldsymbol{Y} = [X, X^2, \ldots, X^k]^{\mathrm{T}}$,
  - $\omega_j = \frac{\partial}{\partial \boldsymbol{\theta}} \mu_j^{-1}(\boldsymbol{\theta})$, $j \in \{1, \ldots, k\}$.

The last statement can be used to find standard errors and (asymptotic) confidence intervals. However, there is an easier way: the (parametric) bootstrap, we'll see!

[1] Notice these properties are less relevant now: the parametric model is <u>usually</u> an approximation to the data generating process: there no "true" $\boldsymbol{\theta}$ to recover from data. See Model Selection for a different perspective.

Method of Moments Estimators

Properties (II)

Method of Moments Estimator (MME)

The MM $\widehat{\boldsymbol{\theta}}_n$, is the value that solve $\mathcal{S}_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n)$

$$\widehat{\boldsymbol{\theta}}_n = \operatorname*{solve}_{\boldsymbol{\theta} \in \Theta} \mathcal{S}_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n).$$

Before we proceed, please appreciate the way in which the sampling process (assuming here a well–specified model) induces variability in our MM estimator:

$$\underbrace{\text{Sample } \boldsymbol{x}_n \text{ from } F_{\boldsymbol{\theta}}(\cdot) \quad \rightsquigarrow \quad \text{Build } \mathcal{S}(\,\cdot\,|\,\boldsymbol{x}_n) \quad \rightsquigarrow \quad \widehat{\boldsymbol{\theta}}_n(\boldsymbol{x}_n) = \operatorname*{solve}_{\boldsymbol{\theta} \in \Theta} \mathcal{S}(\boldsymbol{\theta} \,|\, \boldsymbol{x}_n)}_{\leftsquigarrow \quad \text{repeat} \quad \leftsquigarrow}$$

- The system we're solving is random being a function of the sample $\boldsymbol{X}_n$...
- ...and the MM estimator is random being a zero of this random system...
- ...so its sampling distribution arises from the complex transformation which morphs $\boldsymbol{X}_n$ and its distribution into that of a zero of $\mathcal{S}(\,\cdot\,|\,\boldsymbol{X}_n)$.
- Luckily, as we have seen, somebody has done all the hard work for us!

The Likelihood Way...

...a successful (parametric) story...

Maximum Likelihood Estimators

Definition

The most common method for estimating parameters in a parametric model is the maximum likelihood method.

Likelihood Function

- Let $\boldsymbol{X}_n = \{X_1, \ldots, X_n\}$ be IID with PDF or PMF $f(x \,|\, \boldsymbol{\theta})$.
- Then, the likelihood function is just the joint distribution of the random sample treated as a $\underline{\text{function of the parameter vector}}$:

$$\mathcal{L}_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n) \overset{\text{IID}}{=} \prod_{i=1}^{n} f(X_i \,|\, \boldsymbol{\theta}).$$

- It is often convenient to work with log–likelihood function:

$$\ell_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n) = \log \mathcal{L}_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n) \overset{\text{IID}}{=} \sum_{i=1}^{n} \log f(X_i \,|\, \boldsymbol{\theta}).$$

Maximum Likelihood Estimator (MLE)

The MLE $\widehat{\boldsymbol{\theta}}_n$, is the value that maximizes $\mathcal{L}_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n)$ or, equivalently, $\ell(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n)$:

$$\widehat{\boldsymbol{\theta}}_n = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmax}}\ \mathcal{L}_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n) = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmax}}\ \ell_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n).$$

Maximum Likelihood Estimators

Definition

The most common method for estimating parameters in a parametric model is the maximum likelihood method.

Likelihood Function

- Let $\boldsymbol{X}_n = \{X_1, \ldots, X_n\}$ be IID with PDF or PMF $f(x \,|\, \boldsymbol{\theta})$.
- Then, the likelihood function is just the joint distribution of the random sample treated as a function of the parameter vector:
$$\mathcal{L}_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n) = \prod_{i=1}^{n} f(X_i \,|\, \boldsymbol{\theta}).$$
- It is often convenient to work with the log–likelihood function:
$$\ell_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n) = \log \mathcal{L}_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n) = \sum_{i=1}^{n} \log f(X_i \,|\, \boldsymbol{\theta}).$$

Maximum Likelihood Estimator (MLE)

- In other words, the MLE is the value of the parameter $\boldsymbol{\theta}$ that maximizes the joint probability of observing the sample we actually collected.
- If we multiply the likelihood function by any positive value non depending on $\boldsymbol{\theta}$, then this will not change the MLE. Hence, we shall often drop constants.

Maximum Likelihood Estimators

Example (I)

Bernoulli Likelihood

- Let $\boldsymbol{X}_n = \{X_1, \dots, X_n\}$ be IID $\text{Ber}(p)$, the parameter is $p \in [0,1]$.
- The likelihood and the log–likelihood functions are

$$\begin{aligned}
\mathcal{L}_n(p \,|\, \boldsymbol{X}_n) &= \prod_{i=1}^{n} \text{Ber}(X_i \,|\, p) = \prod_{i=1}^{n} p^{X_i}(1-p)^{1-X_i} = p^{\sum_i X_i}(1-p)^{n-\sum_i X_i} \\
&= p^{n\,\bar{X}_n}(1-p)^{n-n\,\bar{X}_n} \\
\ell_n(p \,|\, \boldsymbol{X}_n) &= n\,\bar{X}_n \log(p) + (n - n\,\bar{X}_n)\log(1-p).
\end{aligned}$$

- Taking the derivative & solving the associated equation we get the MLE:

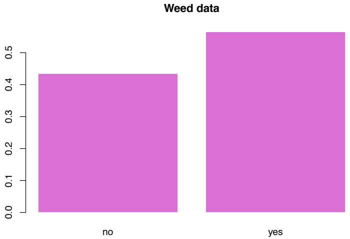
$$\frac{\partial}{\partial p}\ell_n(p \,|\, \boldsymbol{X}_n) = 0 \Leftrightarrow \frac{n\,\bar{X}_n}{p} - \frac{n - n\,\bar{X}_n}{1-p} = 0 \Leftrightarrow (1-p)\bar{X}_n - p(1-\bar{X}_n) = 0$$

$$\Leftrightarrow \bar{X}_n - p = 0 \Leftrightarrow \widehat{p} = \bar{X}_n.$$

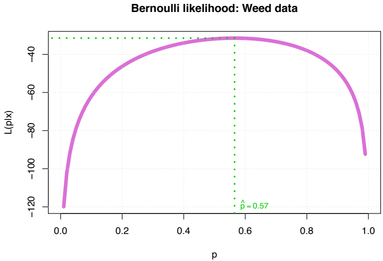
Maximum Likelihood Estimators

Example (I)

```r
# Get the weed data
load("meweed.RData")
n  <- length(meweed)
tt <- prop.table(table(meweed))
barplot(tt, col = "orchid",
  border = "white", main = "Weed data")
p.h <- tt["yes"]; p.hat # MLE

# Bernoulli likelihood
L <- function(p, x){
  n <- length(x)
  x.bar <- mean(x, na.rm = T)
  n*x.bar*log(p) + (n - n*x.bar)*log(1-p)
}

# Transform factor > numeric: "yes" = 1
dd <- as.numeric(as.character(
  factor(meweed, labels = c("0","1")) ))
# Plot
curve(L(x,dd), lwd = 6, col = "orchid",
      xlab = "p", ylab = "L(p|x)")
segments(-5, L(p.h,dd), p.hat, L(p.h,dd),
         lty = 3, lwd = 3, col = "green3")
segments(p.h, -150, p.h, L(p.hat,dd),
         lty = 3, lwd = 3, col = "green3")
text(p.h, -120,
     bquote(hat(p) == .(round(p.h,2))),
     pos = 4, cex = .8, col = "green3")
grid()
```



Weed data



Bernoulli likelihood: Weed data

Maximum Likelihood Estimators

Example (II)

Gaussian Likelihood

- Let $\boldsymbol{X}_n = \{X_1, \ldots, X_n\}$ be IID $\mathsf{N}(\mu, \sigma^2)$, the parameter is $\boldsymbol{\theta} = [\mu, \sigma^2]^{\mathtt{T}}$.
- The likelihood and the log–likelihood functions, ignoring constants, are

$$\begin{aligned}
\mathcal{L}_n(\mu, \sigma \,|\, \boldsymbol{X}_n) &= \prod_{i=1}^{n} \mathsf{N}(X_i \,|\, \mu, \sigma^2) \propto \prod_{i=1}^{n} \frac{1}{\sigma} \exp\left\{-\frac{1}{2\sigma^2}(X_i - \mu \pm \bar{X}_n)^2\right\} \\
&= \prod_{i=1}^{n} \frac{1}{\sigma} \exp\left\{-\frac{1}{2\sigma^2}\left[(X_i - \bar{X}_n) + (\bar{X}_n - \mu)\right]^2\right\} \\
&= \frac{1}{\sigma^n} \exp\left\{-\frac{n\,S_n^2}{2\sigma^2}\right\} \exp\left\{-\frac{n\,(\bar{X}_n - \mu)^2}{2\sigma^2}\right\} \\
\ell_n(\mu, \sigma \,|\, \boldsymbol{X}_n) &= -n\,\log(\sigma) - \frac{n\,S_n^2}{2\sigma^2} - \frac{n\,(\bar{X}_n - \mu)^2}{2\sigma^2}.
\end{aligned}$$

- Solving the system of normal equations

$$\frac{\partial}{\partial \mu} \ell_n(\mu, \sigma \,|\, \boldsymbol{X}_n) = 0 \quad \text{and} \quad \frac{\partial}{\partial \sigma} \ell_n(\mu, \sigma \,|\, \boldsymbol{X}_n) = 0,$$

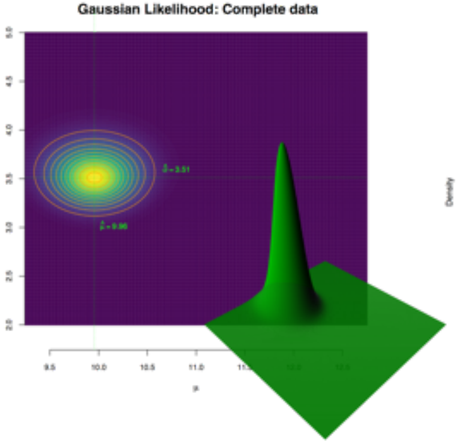
  we get the MLE'S:

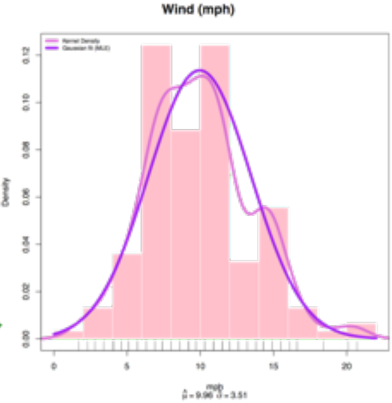
$$\widehat{\mu} = \bar{X}_n \quad \text{and} \quad \widehat{\sigma} = S_n = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X}_n)^2}.$$

Maximum Likelihood Estimators

Example (II)

Gaussian fit to the data density via ML

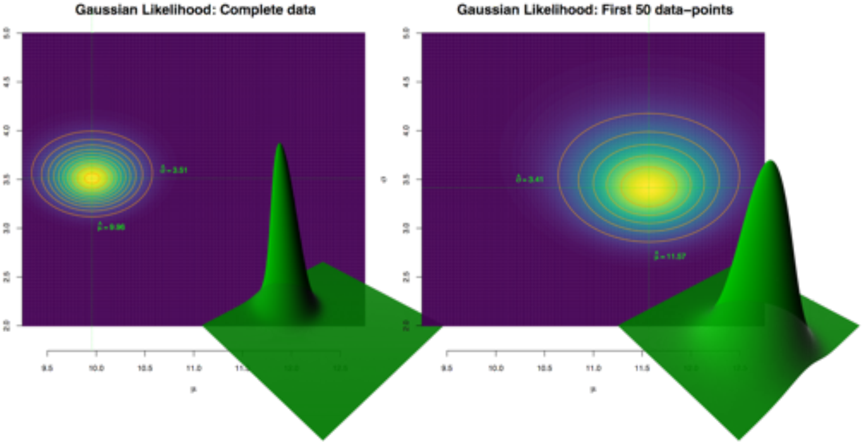




Maximum Likelihood Estimators

Example (II)

The larger the sample size, the more concentrated the likelihood



Maximum Likelihood Estimators

Computational Issues (I)

Maximum Likelihood Estimator (MLE)

The MLE $\widehat{\boldsymbol{\theta}}_n$, is the value that maximizes $\mathcal{L}_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n)$ or, equivalently, $\ell(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n)$:

$$\widehat{\boldsymbol{\theta}}_n = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmax}} \, \mathcal{L}_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n) = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmax}} \, \ell_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n).$$

- As we've seen, in simple (although important) cases, we can find the MLE analytically by implementing the usual first order conditions.
- More often, we need to find it by numerical methods.
- Two useful options are: Newton–Raphson, and the so called EM or Expectation–Maximization algorithm.
- Both are iterative methods that produce a sequence of values $\{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots\}$ that, under suitable conditions, converges to a (local) extreme of the likelihood surface.
- Let's look in some details inside the Newton–Raphson method focusing on the one–dimensional case.

Maximum Likelihood Estimators

Computational Issues (I)

Maximum Likelihood Estimator (MLE)

The MLE $\widehat{\boldsymbol{\theta}}_n$, is the value that maximizes $\mathcal{L}_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n)$ or, equivalently, $\ell(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n)$:

$$\widehat{\boldsymbol{\theta}}_n = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmax}}\ \mathcal{L}_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n) = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmax}}\ \ell_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n).$$

- To introduce Newton–Raphson, let's note that, being a maximizer of the log–likelihood function, the MLE $\widehat{\theta}_n$ is such that $\ell'(\widehat{\theta}_n) = 0$.
- Hence, expanding $\ell'(\cdot)$ up to the 1^{st} order around any given $\theta^\star$, we get

$$0 = \ell'(\widehat{\theta}_n) \approx \ell'(\theta^\star) + \big(\widehat{\theta}_n - \theta^\star\big)\ell''(\theta^\star)$$

- Solving the previous equation for $\widehat{\theta}_n$ we get

$$\widehat{\theta}_n = \theta^\star - \frac{\ell'(\theta^\star)}{\ell''(\theta^\star)}$$

- Starting with an initial values $\widehat{\theta}^{(0)}$, this suggests the following interative scheme:

$$\widehat{\theta}^{(\mathtt{k}+1)} = \widehat{\theta}^{(\mathtt{k})} - \frac{\ell'\big(\widehat{\theta}^{(\mathtt{k})}\big)}{\ell''\big(\widehat{\theta}^{(\mathtt{k})}\big)} \quad \mathtt{k} \in \{1, \ldots, \mathtt{K}_{\mathtt{max}}\}.$$

Maximum Likelihood Estimators

Computational Issues (II)

Newton–Raphson for $1D$ parameters

Starting with an initial values $\widehat{\theta}^{(0)}$, the Newton–Raphson iteration reads:

$$\widehat{\theta}^{(k+1)} = \widehat{\theta}^{(k)} - \frac{\ell'\left(\widehat{\theta}^{(k)}\right)}{\ell''\left(\widehat{\theta}^{(k)}\right)} \quad k \in \{1, \ldots, K_{max}\}.$$

Example: the `optim()` function in `R`

```r
# <optim> is a minimizer so we
# pass -1 x log-likelihood
ll.normal = function(theta, x) {
  # theta = c(mu, sig2)
  mu = theta[1]; sig2 = theta[2]
  n = length(x)
  a1 = -(n/2)*log(2*pi) - (n/2)*log(sig2)
  a2 = -1/(2*sig2)
  y = (x - mu)^2
  ans = a1 + a2*sum(y)
  # return -1 x loglike
  return(-ans)
}

# Initial value
theta.0 = c(5, 5)
```

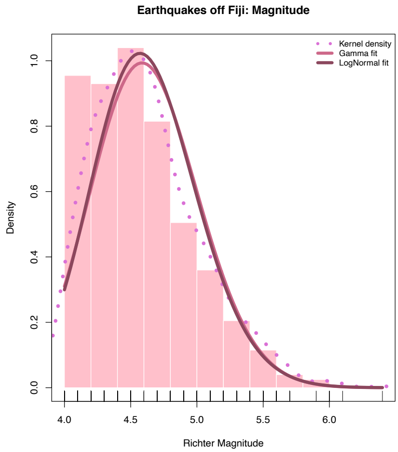
```r
# Compute MLE and approximate SE
# via the numerical Hessian
ans = optim(par = theta.0,
  fn = ll.normal,  x = x,
  method = "BFGS", hessian = T)

names(ans)  # what's inside?
rbind(exact = mle,
  approx = ans$par)
#         mean      var
# exact  9.957516  12.33042
# approx 9.957520  12.33043
ans$hessian
se.mle = sqrt(diag(solve(ans$hessian)))
se.mle
# 0.2838856 1.4097683
```

Maximum Likelihood Estimators

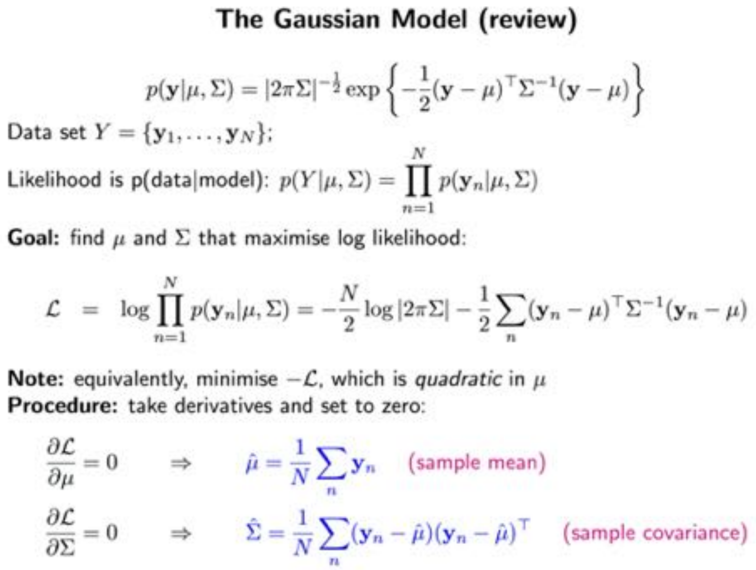
Computational Issues (III)

One interesting option to easily try different standard parametric models is the `fitdistr()` function from the package `MASS`

```r
# Get some data from base R
data("quakes"); ?quakes
x <- quakes$mag # Focus on magnitude

# Take a look
hist(x, prob = T, col = "pink",
  border = "white", main = " ",
  xlab = "Richter Magnitude")
lines(density(x), col = "orchid",
  lty = 3, lwd = 6); rug(x); box()

# Gamma and Lognormal fits (+ SE)
library(MASS)
gfit <- fitdistr(x, "gamma"); gfit
lfit <- fitdistr(x, "lognormal"); lfit
names(gfit); names(lfit)

# Add fit
curve(dgamma(x, gfit$estimate["shape"],
  gfit$estimate["rate"]), lwd = 5,
  col = "palevioletred3", add = T)
curve(dlnorm(x, lfit$estimate["meanlog"],
  lfit$estimate["sdlog"]), lwd = 5,
  col = "palevioletred4", add = T)
```



Maximum Likelihood Estimators



The Gaussian Model (review)

$$p(\mathbf{y}|\mu,\Sigma) = |2\pi\Sigma|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mathbf{y}-\mu)^\top\Sigma^{-1}(\mathbf{y}-\mu)\right\}$$

Data set $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$;

Likelihood is p(data|model): $p(Y|\mu,\Sigma) = \prod_{n=1}^{N} p(\mathbf{y}_n|\mu,\Sigma)$

Goal: find μ and Σ that maximise log likelihood:

$$\mathcal{L} \;=\; \log\prod_{n=1}^{N} p(\mathbf{y}_n|\mu,\Sigma) = -\frac{N}{2}\log|2\pi\Sigma| - \frac{1}{2}\sum_n(\mathbf{y}_n-\mu)^\top\Sigma^{-1}(\mathbf{y}_n-\mu)$$

Note: equivalently, minimise $-\mathcal{L}$, which is *quadratic* in μ

Procedure: take derivatives and set to zero:

$$\frac{\partial\mathcal{L}}{\partial\mu} = 0 \qquad \Rightarrow \qquad \hat{\mu} = \frac{1}{N}\sum_n \mathbf{y}_n \qquad \text{(sample mean)}$$

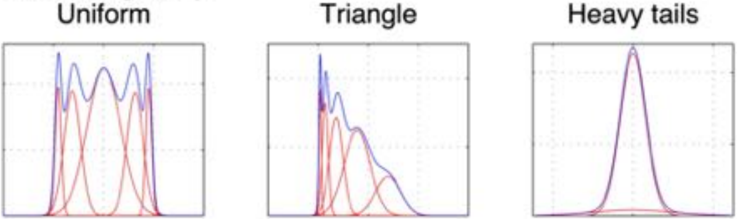
$$\frac{\partial\mathcal{L}}{\partial\Sigma} = 0 \qquad \Rightarrow \qquad \hat{\Sigma} = \frac{1}{N}\sum_n(\mathbf{y}_n-\hat{\mu})(\mathbf{y}_n-\hat{\mu})^\top \qquad \text{(sample covariance)}$$

Maximum Likelihood Estimators

Mixtures of Gaussians – MoG

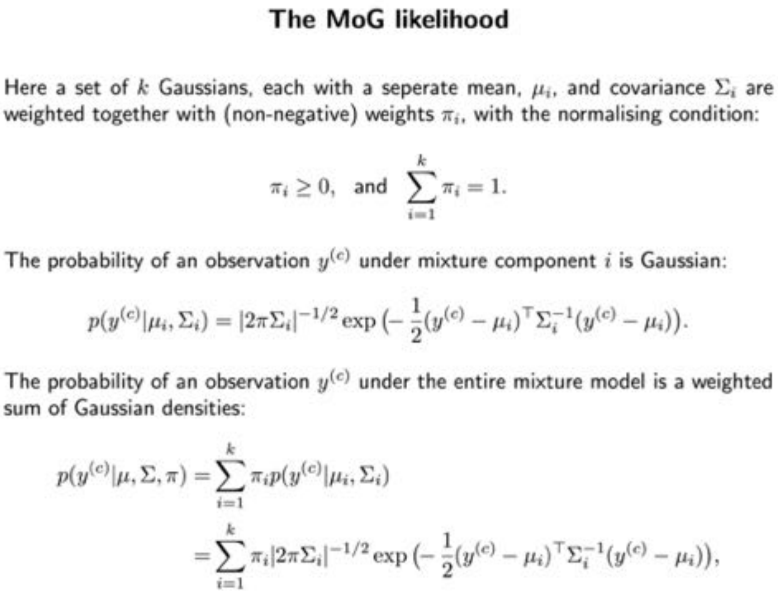
Use probabilistic mixtures of simple (eg Gaussian) density models.

Some examples where non-Gaussian densities are modelled (aproximated) as a mixture of Gaussians. The red curves show the (weighted) Gaussians, and the blue curve the resulting density.



The advantage of this mixture approach, is that given enough mixture components we can model (almost) any density (as accurately as desired), but we still only need to work with the well known Gaussian form.

Maximum Likelihood Estimators



The MoG likelihood

Here a set of k Gaussians, each with a seperate mean, μ_i, and covariance Σ_i are weighted together with (non-negative) weights π_i, with the normalising condition:

$$\pi_i \geq 0, \quad \text{and} \quad \sum_{i=1}^{k} \pi_i = 1.$$

The probability of an observation $y^{(c)}$ under mixture component i is Gaussian:

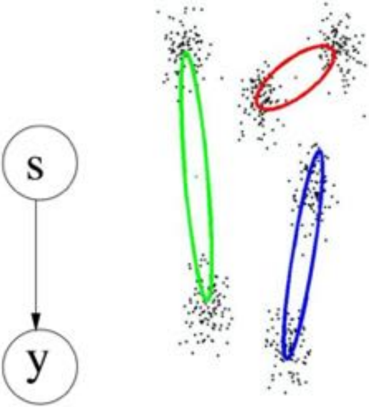
$$p(y^{(c)}|\mu_i, \Sigma_i) = |2\pi\Sigma_i|^{-1/2} \exp\big(-\frac{1}{2}(y^{(c)} - \mu_i)^\top \Sigma_i^{-1}(y^{(c)} - \mu_i)\big).$$

The probability of an observation $y^{(c)}$ under the entire mixture model is a weighted sum of Gaussian densities:

$$\begin{aligned} p(y^{(c)}|\mu, \Sigma, \pi) &= \sum_{i=1}^{k} \pi_i p(y^{(c)}|\mu_i, \Sigma_i) \\ &= \sum_{i=1}^{k} \pi_i |2\pi\Sigma_i|^{-1/2} \exp\big(-\frac{1}{2}(y^{(c)} - \mu_i)^\top \Sigma_i^{-1}(y^{(c)} - \mu_i)\big), \end{aligned}$$

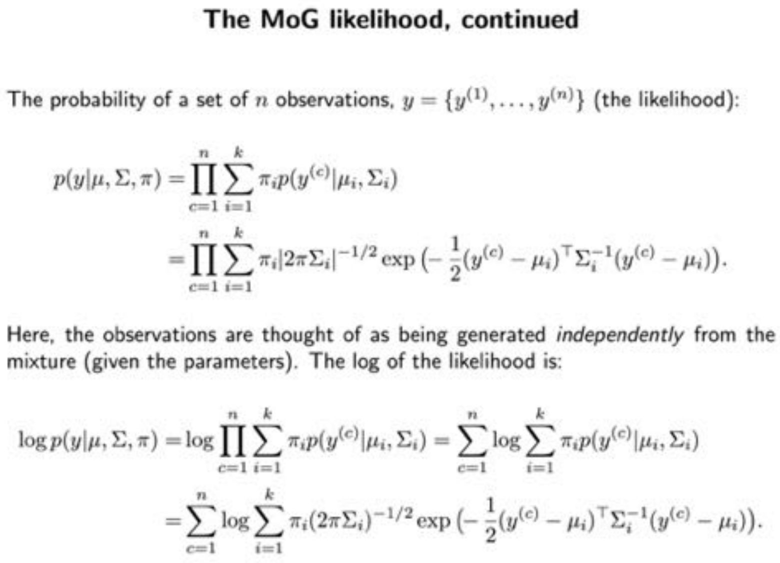
Maximum Likelihood Estimators

MoG as a Latent Variable Model for Clustering



$$P(S^{(c)} = i|\pi) = \pi_i$$
$$p(y^{(c)}|S^{(c)} = i, \mu, \Sigma) = \mathcal{N}(\mu_i, \Sigma_i)$$

Maximum Likelihood Estimators



The MoG likelihood, continued

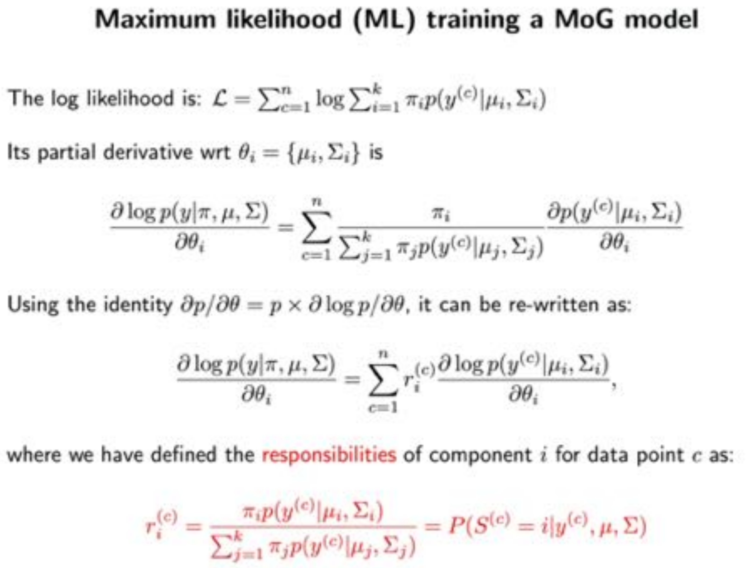
The probability of a set of n observations, $y = \{y^{(1)}, \ldots, y^{(n)}\}$ (the likelihood):

$$\begin{aligned} p(y|\mu, \Sigma, \pi) &= \prod_{c=1}^{n} \sum_{i=1}^{k} \pi_i p(y^{(c)}|\mu_i, \Sigma_i) \\ &= \prod_{c=1}^{n} \sum_{i=1}^{k} \pi_i |2\pi\Sigma_i|^{-1/2} \exp\big(-\frac{1}{2}(y^{(c)} - \mu_i)^\top \Sigma_i^{-1} (y^{(c)} - \mu_i)\big). \end{aligned}$$

Here, the observations are thought of as being generated *independently* from the mixture (given the parameters). The log of the likelihood is:

$$\begin{aligned} \log p(y|\mu, \Sigma, \pi) &= \log \prod_{c=1}^{n} \sum_{i=1}^{k} \pi_i p(y^{(c)}|\mu_i, \Sigma_i) = \sum_{c=1}^{n} \log \sum_{i=1}^{k} \pi_i p(y^{(c)}|\mu_i, \Sigma_i) \\ &= \sum_{c=1}^{n} \log \sum_{i=1}^{k} \pi_i (2\pi\Sigma_i)^{-1/2} \exp\big(-\frac{1}{2}(y^{(c)} - \mu_i)^\top \Sigma_i^{-1} (y^{(c)} - \mu_i)\big). \end{aligned}$$

Maximum Likelihood Estimators



Maximum likelihood (ML) training a MoG model

The log likelihood is: $\mathcal{L} = \sum_{c=1}^{n} \log \sum_{i=1}^{k} \pi_i p(y^{(c)}|\mu_i, \Sigma_i)$

Its partial derivative wrt $\theta_i = \{\mu_i, \Sigma_i\}$ is

$$\frac{\partial \log p(y|\pi, \mu, \Sigma)}{\partial \theta_i} = \sum_{c=1}^{n} \frac{\pi_i}{\sum_{j=1}^{k} \pi_j p(y^{(c)}|\mu_j, \Sigma_j)} \frac{\partial p(y^{(c)}|\mu_i, \Sigma_i)}{\partial \theta_i}$$

Using the identity $\partial p/\partial \theta = p \times \partial \log p/\partial \theta$, it can be re-written as:

$$\frac{\partial \log p(y|\pi, \mu, \Sigma)}{\partial \theta_i} = \sum_{c=1}^{n} r_i^{(c)} \frac{\partial \log p(y^{(c)}|\mu_i, \Sigma_i)}{\partial \theta_i},$$

where we have defined the responsibilities of component i for data point c as:

$$r_i^{(c)} = \frac{\pi_i p(y^{(c)}|\mu_i, \Sigma_i)}{\sum_{j=1}^{k} \pi_j p(y^{(c)}|\mu_j, \Sigma_j)} = P(S^{(c)} = i|y^{(c)}, \mu, \Sigma)$$

Maximum Likelihood Estimators

Derivatives of log likelihood

For the means we get:

$$\frac{\partial \log p(y|\pi, \mu, \Sigma)}{\partial \mu_i} = \sum_{c=1}^{n} r_i^{(c)} \Sigma_i^{-1} (y^{(c)} - \mu_i)$$

and for the *precisions* (inverse variances, Σ_i^{-1}):

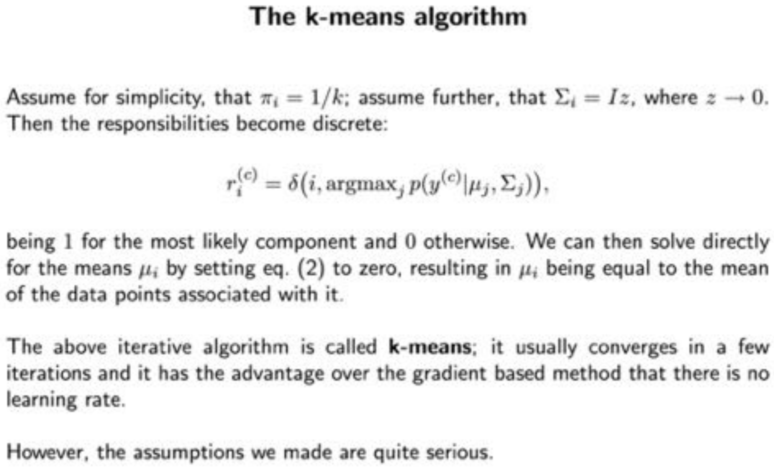
$$\frac{\partial \log p(y|\pi, \mu, \Sigma)}{\Sigma_i^{-1}} = \frac{1}{2} \sum_{c=1}^{n} r_i^{(c)} \left(\Sigma_i - (y^{(c)} - \mu_i)(y^{(c)} - \mu_i)^\top \right).$$

Finally, the partial derivative wrt the mixing proportions is:

$$\frac{\partial \log p(y|\pi, \mu, \Sigma)}{\partial \pi_i} = \sum_{c=1}^{n} \frac{p(y^{(c)}|\mu_i, \Sigma_i)}{\sum_{j=1}^{k} \pi_j p(y^{(c)}|\mu_j, \Sigma_j)}$$

These equations together can be used for gradient based learning; eg taking small steps in the direction of the gradient (or using conjugate gradients).

Maximum Likelihood Estimators



The k-means algorithm

Assume for simplicity, that $\pi_i = 1/k$; assume further, that $\Sigma_i = Iz$, where $z \to 0$. Then the responsibilities become discrete:

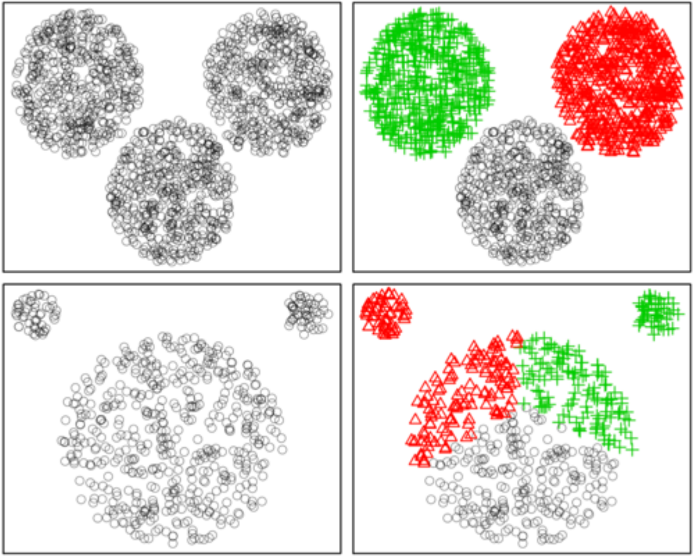
$$r_i^{(c)} = \delta\big(i, \mathrm{argmax}_j\, p(y^{(c)}|\mu_j, \Sigma_j)\big),$$

being 1 for the most likely component and 0 otherwise. We can then solve directly for the means μ_i by setting eq. (2) to zero, resulting in μ_i being equal to the mean of the data points associated with it.

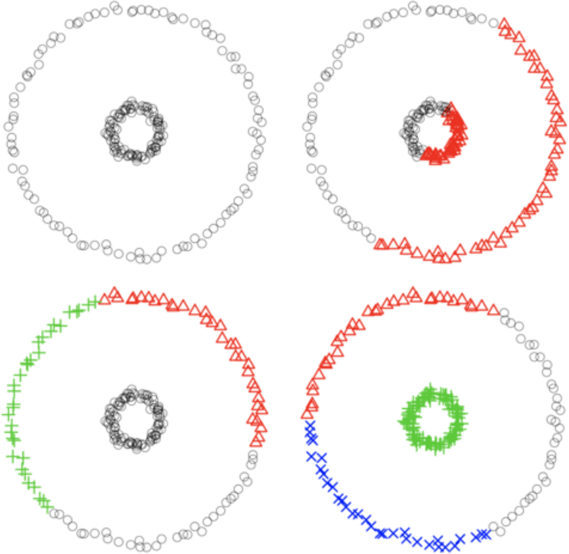
The above iterative algorithm is called **k-means**; it usually converges in a few iterations and it has the advantage over the gradient based method that there is no learning rate.

However, the assumptions we made are quite serious.

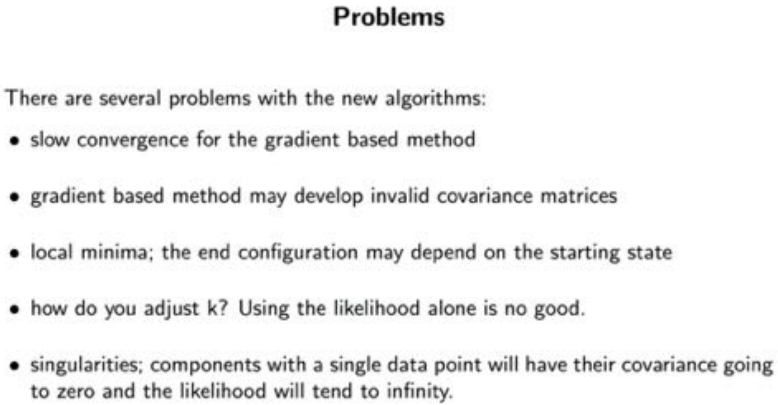
Maximum Likelihood Estimators



Maximum Likelihood Estimators



Maximum Likelihood Estimators

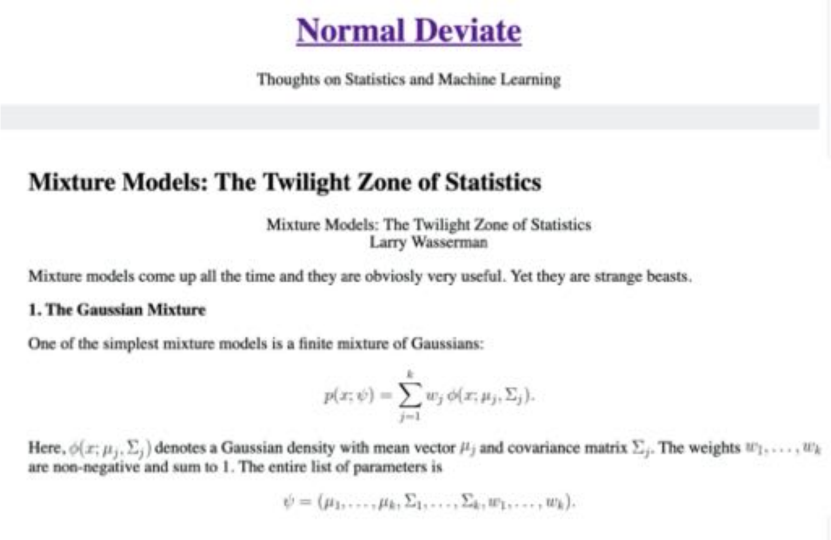


Problems

There are several problems with the new algorithms:

- slow convergence for the gradient based method
- gradient based method may develop invalid covariance matrices
- local minima; the end configuration may depend on the starting state
- how do you adjust k? Using the likelihood alone is no good.
- singularities; components with a single data point will have their covariance going to zero and the likelihood will tend to infinity.

Maximum Likelihood Estimators



Normal Deviate

Thoughts on Statistics and Machine Learning

Mixture Models: The Twilight Zone of Statistics

Mixture Models: The Twilight Zone of Statistics
Larry Wasserman

Mixture models come up all the time and they are obviosly very useful. Yet they are strange beasts.

1. The Gaussian Mixture

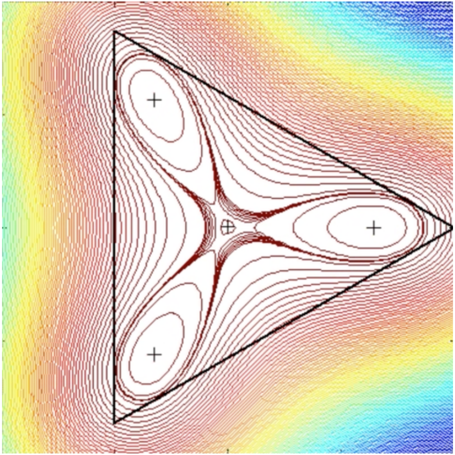
One of the simplest mixture models is a finite mixture of Gaussians:

$$p(x;\psi) = \sum_{j=1}^{k} w_j\, \phi(x;\mu_j,\Sigma_j).$$

Here, $\phi(x;\mu_j,\Sigma_j)$ denotes a Gaussian density with mean vector μ_j and covariance matrix Σ_j. The weights $w_1,\ldots,w_k$ are non-negative and sum to 1. The entire list of parameters is

$$\psi = (\mu_1,\ldots,\mu_k,\Sigma_1,\ldots,\Sigma_k,w_1,\ldots,w_k).$$

Maximum Likelihood Estimators



Maximum Likelihood Estimators

Discrete Comput Geom (2013) 49:797–822
DOI 10.1007/s00454-013-9517-x

Add Isotropic Gaussian Kernels at Own Risk: More and More Resilient Modes in Higher Dimensions

Herbert Edelsbrunner · Brittany Terese Fasy · Günter Rote

Received: 11 June 2012 / Revised: 11 February 2013 / Accept
Published online: 11 June 2013
© Springer Science+Business Media New York 2013

Abstract The fact that a sum of isotropic G
kernels is surprising. Extra (ghost) mode
studied in higher dimensions. We study
which there are exactly $n + 2$ mode
lines, which we call axes, and stud
axes in order to discover that ther
configuration. Although the exi
difficulty of finding examples

arXiv:1702.05066v3 [math.ST] 18 Apr 2019

MAXIMUM NUMBER OF MODES OF GAUSSIAN MIXTURES

CARLOS AMÉNDOLA, ALEXANDER ENGSTRÖM, AND CHRISTIAN HAASE

Abstract. Gaussian mixture models are widely used in Statistics. A fundamental aspect of these distributions is the study of the local maxima of the density, or modes. In particular, it is not known how many modes a mixture of k Gaussians in d dimensions can have. We give a brief account of this problem's history. Then, we give improved lower bounds and the first upper bound on the maximum number of modes, provided it is finite.

1. Introduction

The d-dimensional Gaussian distribution $N(\mu, \Sigma)$ can be defined by its probability density function

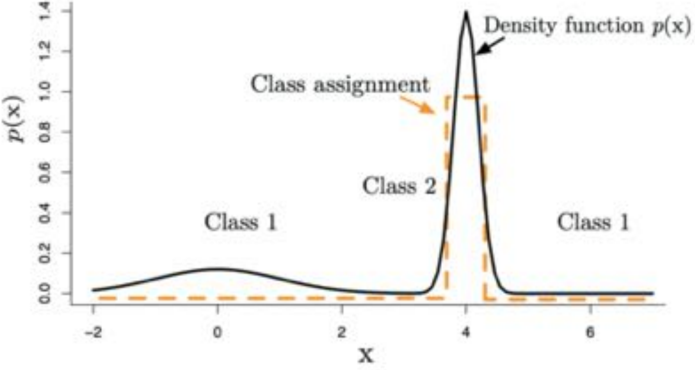
$$\phi(x) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}, \tag{1}$$

where $\mu \in \mathbb{R}^d$ is the mean vector and the symmetric positive definite $d \times d$ matrix Σ i
covariance matrix. This density has a unique absolute maximum at $x = \mu$
a mixture distribution X consisting of k Gaussian components X
weights α_i for $i = 1, \ldots, k$, so that:

(2)

This is agai

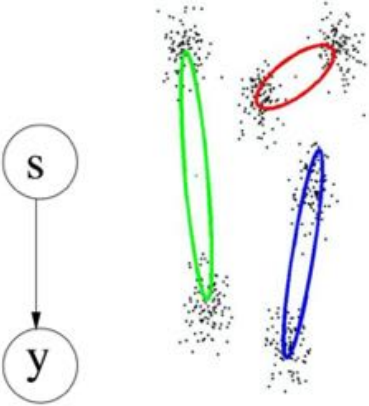
Maximum Likelihood Estimators



Mixtures are used as a parametric method for finding clusters
Observations with $x = 0$ and $x = 6$ are both classified into the first component

Maximum Likelihood Estimators

MoG as a Latent Variable Model for Clustering



$$P(S^{(c)} = i|\pi) = \pi_i$$
$$p(y^{(c)}|S^{(c)} = i, \mu, \Sigma) = \mathcal{N}(\mu_i, \Sigma_i)$$

Maximum Likelihood Estimators

The Expectation Maximization (EM) algorithm

Given a set of observed (visible) variables V, a set of unobserved (hidden / latent / missing) variables H, and model parameters θ, optimize the log likelihood:

$$\mathcal{L}(\theta) = \log p(V|\theta) = \log \int p(H, V|\theta) dH,$$

where we have written the marginal for the visibles in terms of an integral over the joint distribution for hidden and visible variables.

Using *Jensen's inequality* for any distribution of hidden states $q(H)$ we have:

$$\mathcal{L} = \log \int q(H) \frac{p(H, V|\theta)}{q(H)} dH \geq \int q(H) \log \frac{p(H, V|\theta)}{q(H)} \, dH = \mathcal{F}(q, \theta),$$

defining the $\mathcal{F}(q, \theta)$ functional, which is a lower bound on the log likelihood.

In the EM algorithm, we alternately optimize $\mathcal{F}(q, \theta)$ wrt q and θ, and we can prove that this will never decrease $\mathcal{L}$.

Maximum Likelihood Estimators

The E and M steps of EM

The lower bound on the log likelihood:

$$\mathcal{F}(q,\theta) = \int q(H) \log \frac{p(H,V|\theta)}{q(H)} dH = \int q(H) \log p(H,V|\theta) dH + \mathcal{H}(q),$$

where $\mathcal{H}(q) = -\int q(H) \log q(H) dH$ is the entropy of q. We iteratively alternate:

E step: optimize $\mathcal{F}(q,\theta)$ wrt the distribution over hidden variables given the parameters:

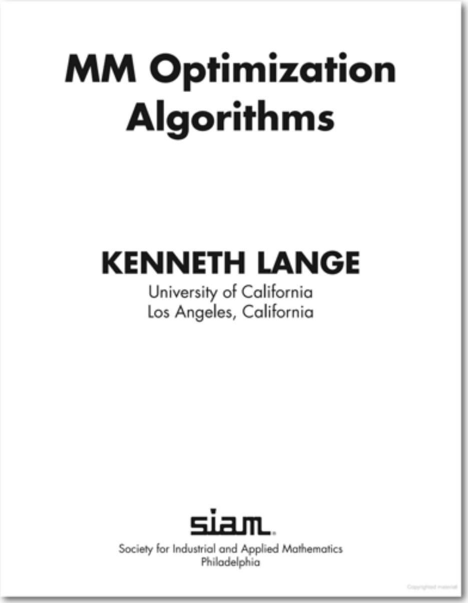
$$q^{(k)}(H) := \underset{q(H)}{\operatorname{argmax}} \ \mathcal{F}\big(q(H), \theta^{(k-1)}\big).$$

M step: maximize $\mathcal{F}(q,\theta)$ wrt the parameters given the hidden distribution:

$$\theta^{(k)} := \underset{\theta}{\operatorname{argmax}} \ \mathcal{F}\big(q^{(k)}(H), \theta\big) = \underset{\theta}{\operatorname{argmax}} \ \int q^{(k)}(H) \log p(H,V|\theta) dH,$$

which is equivalent to optimizing the expected complete-data likelihood $p(H,V|\theta)$, since the entropy of $q(H)$ does not depend on θ.

Maximum Likelihood Estimators



MM Optimization Algorithms

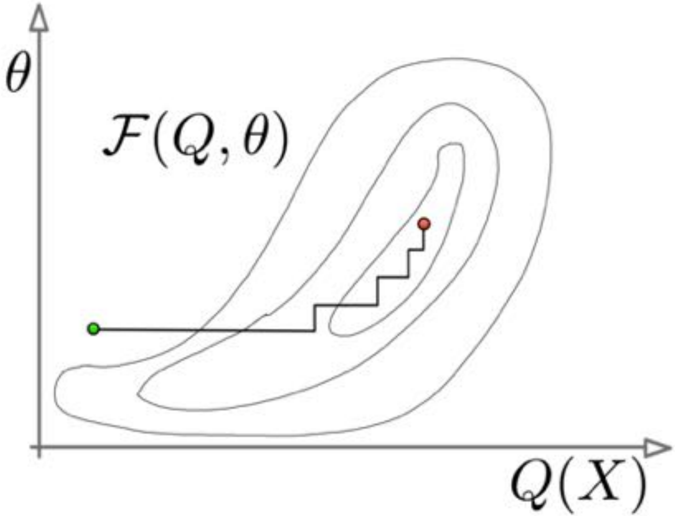
KENNETH LANGE
University of California
Los Angeles, California

siam.
Society for Industrial and Applied Mathematics
Philadelphia

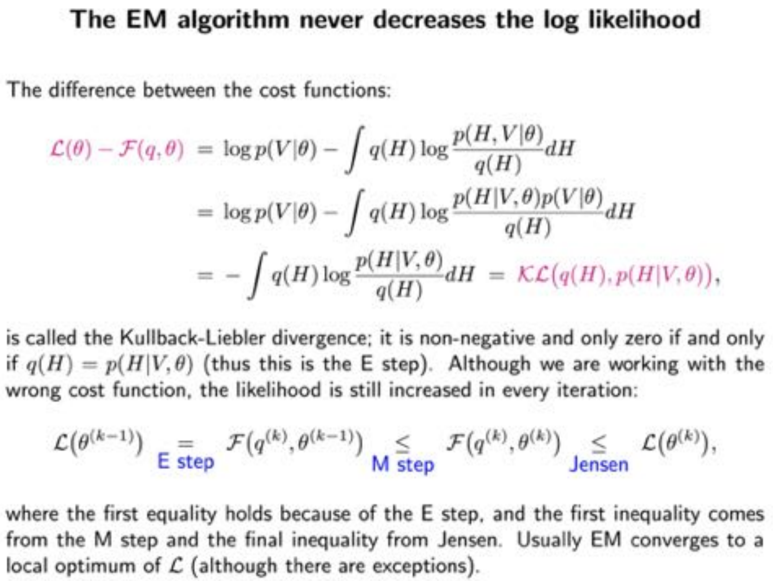
Copyrighted material

Maximum Likelihood Estimators

EM as Coordinate Ascent in $\mathcal{F}$



Maximum Likelihood Estimators



The EM algorithm never decreases the log likelihood

The difference between the cost functions:

$$\begin{aligned}
\mathcal{L}(\theta) - \mathcal{F}(q,\theta) &= \log p(V|\theta) - \int q(H)\log\frac{p(H,V|\theta)}{q(H)}dH \\
&= \log p(V|\theta) - \int q(H)\log\frac{p(H|V,\theta)p(V|\theta)}{q(H)}dH \\
&= -\int q(H)\log\frac{p(H|V,\theta)}{q(H)}dH \;=\; \mathcal{KL}(q(H), p(H|V,\theta)),
\end{aligned}$$

is called the Kullback-Liebler divergence; it is non-negative and only zero if and only if $q(H) = p(H|V,\theta)$ (thus this is the E step). Although we are working with the wrong cost function, the likelihood is still increased in every iteration:

$$\mathcal{L}(\theta^{(k-1)}) \underset{\text{E step}}{=} \mathcal{F}(q^{(k)},\theta^{(k-1)}) \underset{\text{M step}}{\leq} \mathcal{F}(q^{(k)},\theta^{(k)}) \underset{\text{Jensen}}{\leq} \mathcal{L}(\theta^{(k)}),$$

where the first equality holds because of the E step, and the first inequality comes from the M step and the final inequality from Jensen. Usually EM converges to a local optimum of $\mathcal{L}$ (although there are exceptions).

Maximum Likelihood Estimators

The $\mathcal{KL}\big(p(x), q(x)\big)$ is non-negative and zero iff $\forall x:\ p(x) = q(x)$

First let's consider discrete distributions; the Kullback-Liebler divergence is:

$$\mathcal{KL}(p,q) = \sum_i q_i \log \frac{q_i}{p_i}.$$

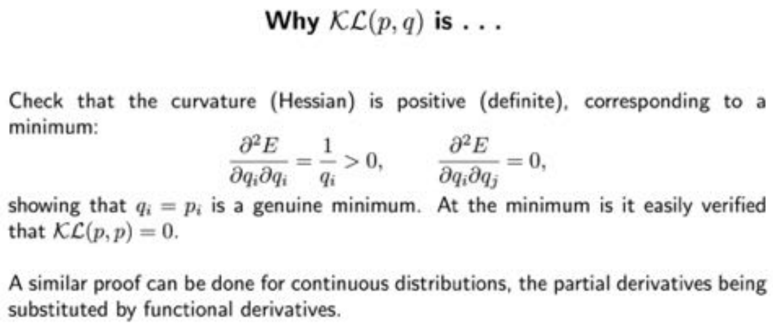
To find the distribution q which minimizes $\mathcal{KL}(p,q)$ we add a lagrange multiplier to enforce the normalization:

$$E = \mathcal{KL}(p,q) + \lambda(1 - \sum_i q_i) = \sum_i q_i \log \frac{q_i}{p_i} + \lambda(1 - \sum_i q_i).$$

We then take partial derives and set to zero:

$$\left.\begin{array}{rcl} \dfrac{\partial E}{\partial q_i} & = & \log(q_i) - \log(p_i) + 1 - \lambda = 0 \Rightarrow q_i = p_i \exp(\lambda - 1) \\ \dfrac{\partial E}{\partial \lambda} & = & 1 - \sum_i q_i = 0 \Rightarrow \sum_i q_i = 1 \end{array}\right\} \Rightarrow q_i = p_i.$$

Maximum Likelihood Estimators



Why $\mathcal{KL}(p,q)$ is . . .

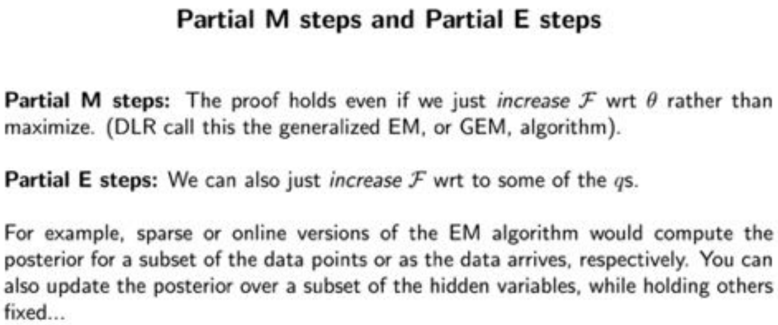
Check that the curvature (Hessian) is positive (definite), corresponding to a minimum:

$$\frac{\partial^2 E}{\partial q_i \partial q_i} = \frac{1}{q_i} > 0, \qquad \frac{\partial^2 E}{\partial q_i \partial q_j} = 0,$$

showing that $q_i = p_i$ is a genuine minimum. At the minimum is it easily verified that $\mathcal{KL}(p,p) = 0$.

A similar proof can be done for continuous distributions, the partial derivatives being substituted by functional derivatives.

Maximum Likelihood Estimators



Partial M steps and Partial E steps

Partial M steps: The proof holds even if we just *increase* $\mathcal{F}$ wrt θ rather than maximize. (DLR call this the generalized EM, or GEM, algorithm).

Partial E steps: We can also just *increase* $\mathcal{F}$ wrt to some of the qs.

For example, sparse or online versions of the EM algorithm would compute the posterior for a subset of the data points or as the data arrives, respectively. You can also update the posterior over a subset of the hidden variables, while holding others fixed...

Maximum Likelihood Estimators

The Gaussian mixture model (E-step)

In the Gaussian mixture density model, the densities of a data point x is:

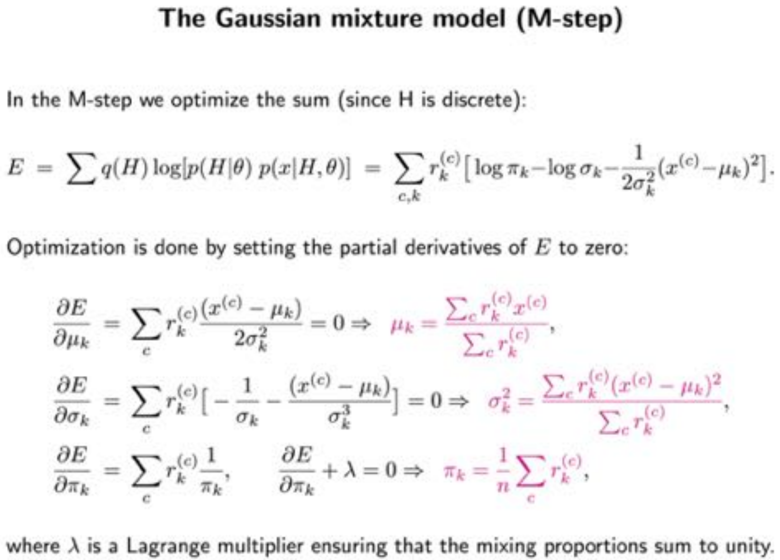
$$p(x|\theta) = \sum_{k=1}^{K} p(H=k|\theta)p(x|H=k,\theta) \propto \sum_{k=1}^{K} \frac{\pi_k}{\sigma_k} \exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2),$$

where θ is the collection of parameters: means μ_k, variances σ_k^2 and mixing proportions $\pi_k = p(H=k|\theta)$.
The hidden variables $H^{(c)}$ indicate which component observation $x^{(c)}$ belongs to.
In the E-step, compute the posterior for $H^{(c)}$ given the current parameters:

$$\begin{aligned} q(H^{(c)}) &= p(H^{(c)}|x^{(c)},\theta) \propto p(x^{(c)}|H^{(c)},\theta)p(H^{(c)}|\theta) \\ r_k^{(c)} \equiv q(H^{(c)}=k) &\propto \frac{\pi_k}{\sigma_k} \exp(-\frac{1}{2\sigma_k^2}(x^{(c)}-\mu_k)^2) \quad \text{(responsibilities)} \end{aligned}$$

with the normalization such that $\sum_k r_k^{(c)} = 1$.

Maximum Likelihood Estimators



The Gaussian mixture model (M-step)

In the M-step we optimize the sum (since H is discrete):

$$E = \sum q(H) \log[p(H|\theta)\; p(x|H,\theta)] = \sum_{c,k} r_k^{(c)} \big[\log \pi_k - \log \sigma_k - \frac{1}{2\sigma_k^2}(x^{(c)} - \mu_k)^2 \big].$$

Optimization is done by setting the partial derivatives of E to zero:

$$\frac{\partial E}{\partial \mu_k} = \sum_c r_k^{(c)} \frac{(x^{(c)} - \mu_k)}{2\sigma_k^2} = 0 \Rightarrow \quad \mu_k = \frac{\sum_c r_k^{(c)} x^{(c)}}{\sum_c r_k^{(c)}},$$

$$\frac{\partial E}{\partial \sigma_k} = \sum_c r_k^{(c)} \big[- \frac{1}{\sigma_k} - \frac{(x^{(c)} - \mu_k)}{\sigma_k^3} \big] = 0 \Rightarrow \quad \sigma_k^2 = \frac{\sum_c r_k^{(c)} (x^{(c)} - \mu_k)^2}{\sum_c r_k^{(c)}},$$

$$\frac{\partial E}{\partial \pi_k} = \sum_c r_k^{(c)} \frac{1}{\pi_k}, \qquad \frac{\partial E}{\partial \pi_k} + \lambda = 0 \Rightarrow \quad \pi_k = \frac{1}{n} \sum_c r_k^{(c)},$$

where λ is a Lagrange multiplier ensuring that the mixing proportions sum to unity.

Maximum Likelihood Estimators

mixtools: An R Package for Analyzing Finite Mixture Models

Tatiana Benaglia
Pennsylvania State University

Didier Chauveau
Université d'Orléans

David R. Hunter
Pennsylvania State University

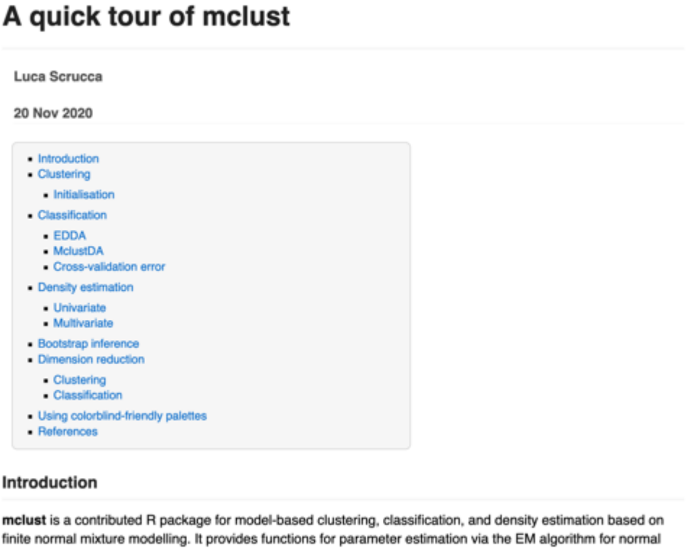
Derek S. Young
Pennsylvania State University

Abstract

The **mixtools** package for R provides a set of functions for analyzing a variety of finite mixture models. These functions include both traditional methods, such as EM algorithms for univariate and multivariate normal mixtures, and newer methods that reflect some recent research in finite mixture models. In the latter category, **mixtools** provides algorithms for estimating parameters in a wide range of different mixture-of-regression contexts, in multinomial mixtures such as those arising from discretizing continuous multivariate data, in nonparametric situations where the multivariate component densities are completely unspecified, and in semiparametric situations such as a univariate location mixture of symmetric but otherwise unspecified densities. Many of the algorithms of the **mixtools** package are EM algorithms or are based on EM-like ideas, so this article includes an overview of EM algorithms for finite mixture models.

Keywords: cutpoint, EM algorithm, mixture of regressions, model-based clustering, nonparametric mixture, semiparametric mixture, unsupervised clustering.

Maximum Likelihood Estimators



A quick tour of mclust

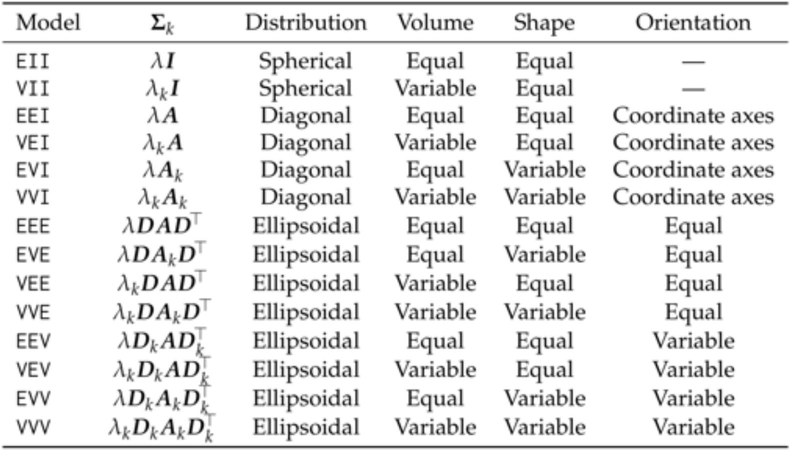
Luca Scrucca

20 Nov 2020

Introduction

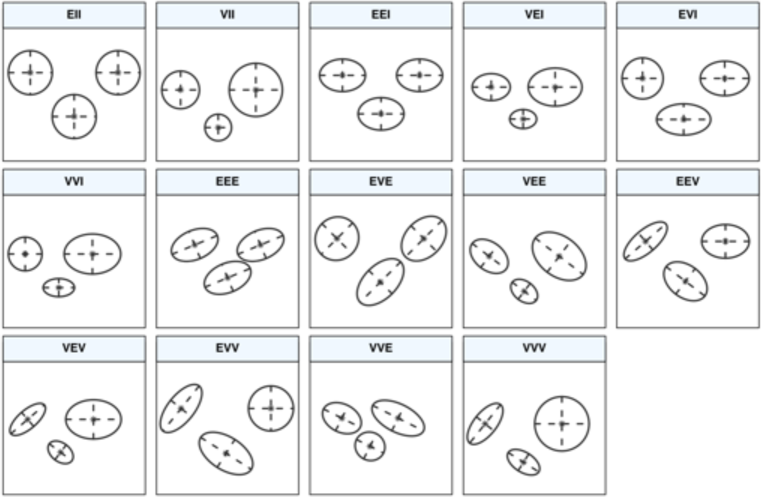
mclust is a contributed R package for model-based clustering, classification, and density estimation based on finite normal mixture modelling. It provides functions for parameter estimation via the EM algorithm for normal

Maximum Likelihood Estimators



Model	Σ_k	Distribution	Volume	Shape	Orientation
EII	$\lambda \boldsymbol{I}$	Spherical	Equal	Equal	—
VII	$\lambda_k \boldsymbol{I}$	Spherical	Variable	Equal	—
EEI	$\lambda \boldsymbol{A}$	Diagonal	Equal	Equal	Coordinate axes
VEI	$\lambda_k \boldsymbol{A}$	Diagonal	Variable	Equal	Coordinate axes
EVI	$\lambda \boldsymbol{A}_k$	Diagonal	Equal	Variable	Coordinate axes
VVI	$\lambda_k \boldsymbol{A}_k$	Diagonal	Variable	Variable	Coordinate axes
EEE	$\lambda \boldsymbol{D} \boldsymbol{A} \boldsymbol{D}^\top$	Ellipsoidal	Equal	Equal	Equal
EVE	$\lambda \boldsymbol{D} \boldsymbol{A}_k \boldsymbol{D}^\top$	Ellipsoidal	Equal	Variable	Equal
VEE	$\lambda_k \boldsymbol{D} \boldsymbol{A} \boldsymbol{D}^\top$	Ellipsoidal	Variable	Equal	Equal
VVE	$\lambda_k \boldsymbol{D} \boldsymbol{A}_k \boldsymbol{D}^\top$	Ellipsoidal	Variable	Variable	Equal
EEV	$\lambda \boldsymbol{D}_k \boldsymbol{A} \boldsymbol{D}_k^\top$	Ellipsoidal	Equal	Equal	Variable
VEV	$\lambda_k \boldsymbol{D}_k \boldsymbol{A} \boldsymbol{D}_k^\top$	Ellipsoidal	Variable	Equal	Variable
EVV	$\lambda \boldsymbol{D}_k \boldsymbol{A}_k \boldsymbol{D}_k^\top$	Ellipsoidal	Equal	Variable	Variable
VVV	$\lambda_k \boldsymbol{D}_k \boldsymbol{A}_k \boldsymbol{D}_k^\top$	Ellipsoidal	Variable	Variable	Variable

Maximum Likelihood Estimators



Maximum Likelihood Estimators

Properties (I)

Maximum Likelihood Estimator (MLE)

The MLE $\widehat{\boldsymbol{\theta}}_n$, is the value that maximizes $\mathcal{L}_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n)$ or, equivalently, $\ell(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n)$:

$$\widehat{\boldsymbol{\theta}}_n = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmax}}\ \mathcal{L}_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n) = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmax}}\ \ell_n(\boldsymbol{\theta} \,|\, \boldsymbol{X}_n).$$

- As with any other statistical procedure, we are interested in evaluating the frequentist properties of the estimators the ML techniques spits out.
- Notice that, although we are living in a simple, constrained parametric world, the MLE, in its full generality, is a pretty complex stochastic object being the maximum of a random objective function.
- As we have seen, sometimes this optimization can be carried out explicitly so that the resulting MLE can be studied directly (i.e. forgetting its tricky genesis). In all the other more intricate situations, we need more general tools to get theoretical guarantees on the MLE performances.
- The complete (asymptotic) theory for the MLE is all above the level of this course, so we give a non-theoretical explanation.

Maximum Likelihood Estimators

Properties (I)

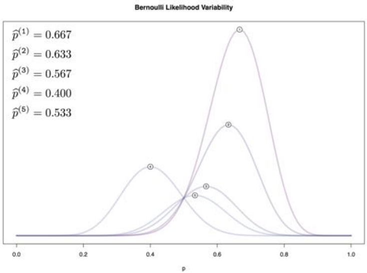
Maximum Likelihood Estimator (MLE)

The MLE $\widehat{\boldsymbol{\theta}}_n$, is the value that maximizes $\mathcal{L}_n(\boldsymbol{\theta} \mid \boldsymbol{X}_n)$ or, equivalently, $\ell(\boldsymbol{\theta} \mid \boldsymbol{X}_n)$:

$$\widehat{\boldsymbol{\theta}}_n = \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} \mathcal{L}_n(\boldsymbol{\theta} \mid \boldsymbol{X}_n) = \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} \ell_n(\boldsymbol{\theta} \mid \boldsymbol{X}_n).$$

Before we proceed, please appreciate once more the (complex!) way in which the sampling process induces variability in our ML estimator:

$$\underbrace{\text{Sample } \boldsymbol{x}_n \text{ from } F_{\boldsymbol{\theta}}(\cdot) \quad \rightsquigarrow \quad \text{Build } \mathcal{L}(\,\cdot \mid \boldsymbol{x}_n) \quad \rightsquigarrow \quad \widehat{\boldsymbol{\theta}}_n(\boldsymbol{x}_n) = \operatorname*{argmax}_{\boldsymbol{\theta}\in\Theta} \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{x}_n)}_{\leftsquigarrow \quad \text{repeat} \quad \leftsquigarrow}$$



Maximum Likelihood Estimators

Properties (II)

Summary

- Under suitable conditions on the model $f(x \mid \theta)$, the maximum likelihood estimators (MLE) sports many appealing properties.
- In sufficiently complicated or even nonparametric problems these conditions will no longer hold and the MLE will no longer be so good.
- Similarly to the method–of–moments estimators, most of these results still hold (with few adjustments) in the misspecified model case.
- In this part we focus only on parametric models where the MLE works well. So here's a summary of the relevant (to us) properties:
  1. The MLE is consistent.
  2. The MLE is asymptotically Normal and the estimated standard error $\widehat{\mathsf{se}}$ can often be computed analytically.
  3. The MLE is equivariant: let $\widehat{\theta}_n$ be the MLE for θ and assume the parameter of interest is $\tau = h(\theta)$. Then the MLE for τ is simply $h(\widehat{\theta}_n)$. In other words, we don't need to *reparametrize*[2] our model in terms of τ and optimize the associated likelihood function.
  4. The MLE is asymptotically efficient: roughly, this means that among all *good* estimators, the MLE has asymptotically the smallest variance.

[2] In general, not all parameterization of a model are born equal: some allow for easier estimation than others.

Maximum Likelihood Estimators

Properties (III)

Achtung Achtung!

Before going ahead, let's recap what we know about the likelihood function:

1. The likelihood function is a function of θ.
2. The likelihood function is not a PDF or a PMF.
3. If the data are IID then the likelihood is

   $$\mathcal{L}(\theta \mid \boldsymbol{X}_n) = \prod_{i=1}^{n} f(x_i \mid \theta).$$

   For dependent data (e.g. space–time series), the likelihood function is still well defined in terms of the joint model but does not (usually) have such a neat face!
4. The likelihood is only defined up to a constant of proportionality. In other words, it is an *equivalence class* of functions ⇝ scale it as you wish for computational convenience!
5. The likelihood function is mainly used...
   - 5.1 ...to generate estimators (the maximum likelihood estimator);
   - 5.2 ...as a key ingredient in Bayesian inference.
6. The likelihood is central to parametric inference but most of its nice statistical properties vanish going nonparametrics (see Appendix A instead, for a success!).

Parametric Confidence Intervals (I)

MLE asymptotics $+\ \delta$–method

Maximum Likelihood Estimators

Properties (IV)

To get *approximate* standard errors and, in the end, confidence sets for the parameter of interest, we may follow different paths.

One standard error, many strategies

- Nonparametric model:
  1. Analytic solution via influence functions.
  2. Computational solution via Nonparametric Bootstrap.
- Parametric model:
  1. Analytic solution via Fisher information.
  2. Computational solution via Parametric Bootstrap.

Maximum Likelihood Estimators

Properties (IV)

To get *approximate* standard errors and, in the end, confidence sets for the parameter of interest, we may follow different paths.

One standard error, many strategies

- Nonparametric model:
  1. Analytic solution via influence functions.
  2. Computational solution via Nonparametric Bootstrap.
- Parametric model:
  1. Analytic solution via Fisher information.
  2. Computational solution via Parametric Bootstrap.

Maximum Likelihood Estimators

Properties (IV)

To get *approximate* standard errors and, in the end, confidence sets for the parameter of interest, we may follow different paths.

One standard error, many strategies

- Nonparametric model:
  1. Analytic solution via influence functions.
  2. Analytic solution via influence functions.
  3. Computational solution via Nonparametric Bootstrap.
- Parametric model:
  1. Analytic solution via Fisher information.
  2. Computational solution via Parametric Bootstrap.

To simplify the treatment, in the following...

1. ...we will first analyze the uni–parametric case $\Theta \subseteq \mathbb{R}$;
2. ...we will assume the (parametric) model is well–specified.

Maximum Likelihood Estimators

Properties (V)

Asymptotic Normality (I)

- It turns out that the distribution of the MLE is asymptotically Normal and we can compute analytically its estimated standard error $\widehat{\mathrm{se}}$. We first need some definitions:

1. If $X \sim f(x|\theta)$, the score function is defined as

$$s(X \mid \theta) = \frac{\partial}{\partial\theta} \log f(X \mid \theta).$$

2. If $\{X_1, \ldots, X_n\}$ are IID as $f(x \mid \theta)$, then the Fisher information is defined to be

$$\mathcal{I}_n(\theta) = \mathbb{V}\mathrm{ar}\left(\sum_{i=1}^{n} s(X_i \mid \theta)\right) \overset{\mathrm{IID}}{=} \sum_{i=1}^{n} \mathbb{V}\mathrm{ar}\big(s(X_i \mid \theta)\big).$$

For $n = 1$ we will write $\mathcal{I}(\theta) = \mathbb{V}\mathrm{ar}\big(s(X \mid \theta)\big)$ instead of $\mathcal{I}_1(\theta)$.

Maximum Likelihood Estimators

Properties (VI)

Asymptotic Normality (II)

- Before we go to the actual result, for practical applications it is useful to highlight a nice simplification (...although at first sight it doesn't seem so...) in the evaluation of $\mathcal{I}_n(\theta)$, that is:

Theorem: For an IID samples $\mathcal{I}_n(\theta) = n \cdot \mathcal{I}(\theta)$ with $\mathcal{I}(\theta) = \mathbb{V}\mathrm{ar}\big(s(X \mid \theta)\big)$. In addition, under suitable conditions, we have

$$\mathcal{I}(\theta) = -\mathbb{E}\left(\frac{\partial}{\partial\theta} s(X \mid \theta)\right) = -\mathbb{E}\left(\frac{\partial^2}{\partial\theta^2} \log f(X \mid \theta)\right)$$

$$= -\int \left(\frac{\partial^2}{\partial\theta^2} \log f(x \mid \theta)\right) f(x \mid \theta) \mathrm{d}\theta.$$

- So, instead of evaluating the variance of a 1^{st} derivative, we just need to get the expected value of the 2^{nd} derivative: usually easier to work out.
- Notice that, being a 2^{nd} derivative, the Fisher information may be interpreted geometrically as the curvature of the log–likelihood around θ: the *higher* the curvature the *more informative* the log–likelihood is.

Maximum Likelihood Estimators

Properties (VI)

Asymptotic Normality (II)

- Before we go to the actual result, for practical applications it is useful to highlight a nice simplification (...although at first sight it doesn't seem so...) in the evaluation of $\mathcal{I}_n(\theta)$, that is:

Theorem: For an IID samples $\mathcal{I}_n(\theta) = n \cdot \mathcal{I}(\theta)$ with $\mathcal{I}(\theta) = \mathbb{V}\mathrm{ar}\big(s(X \mid \theta)\big)$. In addition, under suitable conditions, we have

$$\mathcal{I}(\theta) = -\mathbb{E}\left(\frac{\partial}{\partial\theta} s(X \mid \theta)\right) = -\mathbb{E}\left(\frac{\partial^2}{\partial\theta^2} \log f(X \mid \theta)\right)$$
$$= -\int \left(\frac{\partial^2}{\partial\theta^2} \log f(x \mid \theta)\right) f(x \mid \theta) \mathrm{d}\theta.$$

- Notice also that, in general, the Fisher information will **depend** on the **unknown** parameter θ.
- Finally, we are now in position to state the main result regarding the *asymptotic normality* of the MLE.

Maximum Likelihood Estimators

Properties (VII)

Asymptotic Normality (III)

Theorem: Let $\widehat{\theta}_n$ denotes the MLE for θ and $\mathrm{se} = \sqrt{\mathbb{V}\mathrm{ar}(\widehat{\theta}_n)}$. Then, under suitable conditions, the following hold:

1. For large samples, the sampling distribution of $\widehat{\theta}_n$ can be *approximated* by a Normal with mean θ and variance se^2. In addition, the asymptotic se can be *approximated* by:

$$\mathrm{se} \approx \left[\mathcal{I}_n(\theta)\right]^{-\frac{1}{2}}$$

[High curvature $\leftrightsquigarrow$ A lot of "information" $\leftrightsquigarrow$ Smaller standard error]

2. Since $\mathcal{I}_n(\theta)$ depends on the unknown θ, the proposed approximation to the se seems to be useless. Anyway, it can be shown that the previous result still holds if we **estimate** the se by plugging $\widehat{\theta}_n$ in the definition of $\mathcal{I}_n(\theta)$. That is, $\widehat{\theta}_n \approx \mathsf{N}(\theta, \widehat{\mathrm{se}})$ where

$$\widehat{\mathrm{se}} \approx \left[\mathcal{I}_n(\widehat{\theta}_n)\right]^{-\frac{1}{2}}$$

Maximum Likelihood Estimators

Properties (VIII)

Example: Bernoulli model

- Let $\{X_1, \ldots, X_n\}$ be IID $\text{Ber}(p)$, so $f(x \mid p) = p^x(1-p)^{1-x}$ with $x \in \{0, 1\}$.
- We already know that the MLE for p is $\widehat{p}_n = \bar{X}_n$, the sample average. So:

$$
\begin{aligned}
\log f(x \mid p) &= x \log p + (1-x)\log(1-p) \\
s(X \mid p) &= \frac{\partial}{\partial p}\log f(X \mid p) = \frac{X}{p} - \frac{1-X}{1-p} \\
-\frac{\partial}{\partial p}s(X \mid p) &= \frac{X}{p^2} + \frac{1-X}{(1-p)^2} \\
\mathcal{I}(p) &= \mathbb{E}\left(-\frac{\partial}{\partial p}s(X \mid p)\right) = \frac{p}{p^2} + \frac{1-p}{(1-p)^2} = \frac{1}{p(1-p)}
\end{aligned}
$$

Thus

$$
\widehat{\text{se}} = \left[\mathcal{I}_n(\widehat{p}_n)\right]^{-\frac{1}{2}} = \left[n \cdot \mathcal{I}(\widehat{p}_n)\right]^{-\frac{1}{2}} = \sqrt{\frac{\widehat{p}_n(1-\widehat{p}_n)}{n}}.
$$

So finally, for large n, we can say that $\widehat{p}_n$ is Normal with mean p and the (estimated) standard error above: pretty easy now to get approximate CI for p...

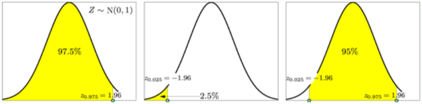
Normal Confidence Intervals

Recap

$$\mathbb{P}_{F_X}\big(\theta \in \mathrm{C}_n(\alpha)\big) \geqslant 1-\alpha \quad \underline{\text{uniformly}} \text{ over } \theta \text{ (if possible)}$$

Basic example: Normal intervals for the mean (asymptotic pivot)

- Let $\widehat{\theta}_n$ be (at least asymptotically) Normal, with $\widehat{\theta}_n \overset{\cdot}{\sim} \mathsf{N}(\theta, \widehat{\mathrm{se}}^2)$.
- Let $z_{\frac{\alpha}{2}} = -\mathsf{qnorm}(\frac{\alpha}{2}) \overset{\text{sym}}{=} \mathsf{qnorm}\big(1-(\frac{\alpha}{2})\big)$.



- Then (at least asymptotically) standardizing $\widehat{\theta}_n$ we get a random variable whose definition depends on the unknown parameter θ, but its distribution does **not**:

$$Z = \frac{\widehat{\theta}_n - \theta}{\widehat{\mathrm{se}}(\widehat{\theta}_n)} \overset{\cdot}{\sim} \mathsf{N}(0,1) \quad \Rightarrow \quad \mathbb{P}\big(|\,Z\,| \leqslant z_{\frac{\alpha}{2}}\big) \overset{\cdot}{\approx} 1-\alpha \quad \text{so, ``pivoting'' on } \theta\ldots$$

$$1-\alpha \overset{\cdot}{\approx} \mathbb{P}\left(\left|\frac{\widehat{\theta}_n-\theta}{\widehat{\mathrm{se}}(\widehat{\theta}_n)}\right| \leqslant z_{\frac{\alpha}{2}}\right) = \mathbb{P}\Big(-\widehat{\theta}_n - z_{\frac{\alpha}{2}}\widehat{\mathrm{se}}(\widehat{\theta}_n) \leqslant -\theta \leqslant -\widehat{\theta}_n + z_{\frac{\alpha}{2}}\widehat{\mathrm{se}}(\widehat{\theta}_n)\Big)$$

$$= \mathbb{P}\Big(\widehat{\theta}_n - z_{\frac{\alpha}{2}}\widehat{\mathrm{se}}(\widehat{\theta}_n) \leqslant \theta \leqslant \widehat{\theta}_n + z_{\frac{\alpha}{2}}\widehat{\mathrm{se}}(\widehat{\theta}_n)\Big)$$

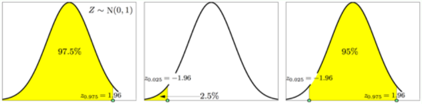
Normal Confidence Intervals

Recap

$$\mathbb{P}_{F_X}(\theta \in \mathrm{C}_n(\alpha)) \geqslant 1-\alpha \quad \underline{\text{uniformly}} \text{ over } \theta \text{ (if possible)}$$

Basic example: Normal intervals for the mean (asymptotic pivot)

- Let $\widehat{\theta}_n$ be (at least asymptotically) Normal, with $\widehat{\theta}_n \dot{\sim} \mathrm{N}(\theta, \widehat{\mathrm{se}}^2)$.
- Let $z_{\frac{\alpha}{2}} = -\mathrm{qnorm}(\frac{\alpha}{2}) \overset{\text{sym}}{=} \mathrm{qnorm}\big(1-(\frac{\alpha}{2})\big)$.



- Then (at least asymptotically) standardizing $\widehat{\theta}_n$ we get a random variable whose definition depends on the unknown parameter θ, but its distribution does **not**:

$$Z = \frac{\widehat{\theta}_n - \theta}{\widehat{\mathrm{se}}(\widehat{\theta}_n)} \dot{\sim} \mathrm{N}(0,1) \quad \Rightarrow \quad \mathbb{P}\big(|\,Z\,| \leqslant z_{\frac{\alpha}{2}}\big) \dot{\approx} 1-\alpha \quad \text{so, “pivoting” on } \theta\ldots$$

- ...we have just proved that, at least asymptotically,

$$\mathrm{C}_n(\alpha) = \Big[\widehat{\theta}_n - z_{\frac{\alpha}{2}}\widehat{\mathrm{se}}(\widehat{\theta}_n), \widehat{\theta}_n + z_{\frac{\alpha}{2}}\widehat{\mathrm{se}}(\widehat{\theta}_n)\Big],$$

  is an *approximate* $(1-\alpha)$ confidence interval for θ.

Normal Confidence Intervals

The δ–method (I)

- The previous machinery provides a simple recipe to get Normal confidence intervals for the parameter of interest θ.
- When our attention moves to $\tau = h(\theta)$ where $h(\cdot)$ is a smooth function, we can rely on the δ–method to make inference on τ.
- First of all notice that, if $\widehat{\theta}_n$ denotes the MLE for θ, then by the equivariance of ML estimators we have that $\widehat{\tau}_n = h(\widehat{\theta}_n)$ is the MLE for τ.

δ–Method

If $\tau = h(\theta)$ where $h(\cdot)$ is differentiable and $h'(\theta) \neq 0$, then

$$\frac{\widehat{\tau}_n - \tau}{\widehat{\mathrm{se}}(\widehat{\tau}_n)} \rightsquigarrow \mathrm{N}(0,1) \quad \text{where} \quad \widehat{\mathrm{se}}(\widehat{\tau}_n) = \left|h'(\widehat{\theta}_n)\right| \widehat{\mathrm{se}}(\widehat{\theta}_n).$$

Hence the following is an asymptotic $(1-\alpha)$ Normal confidence interval for τ:

$$\left(\widehat{\tau}_n - z_{\frac{\alpha}{2}}\widehat{\mathrm{se}}(\widehat{\tau}_n), \widehat{\tau}_n + z_{\frac{\alpha}{2}}\widehat{\mathrm{se}}(\widehat{\tau}_n)\right)$$

Normal Confidence Intervals

The δ–method (II)

δ–Method: $(1-\alpha)$ CI's

$$\left(\widehat{\tau}_n - z_{\frac{\alpha}{2}}\,\widehat{\mathrm{se}}(\widehat{\tau}_n), \widehat{\tau}_n + z_{\frac{\alpha}{2}}\,\widehat{\mathrm{se}}(\widehat{\tau}_n)\right) \quad \text{where} \quad \widehat{\mathrm{se}}(\widehat{\tau}_n) = \left|h'(\widehat{\theta}_n)\right|\widehat{\mathrm{se}}(\widehat{\theta}_n)$$

Example

- Let $\{X_1, \ldots, X_n\}$ be IID as $\mathrm{Ber}(p)$, and let $\tau = h(p) = \log\left(\frac{p}{1-p}\right)$.
- From the previous example, we already know that the MLE for p is $\widehat{p}_n = \bar{X}_n$ and its asymptotic standard error can be approximated by:

$$\widehat{\mathrm{se}}(\widehat{p}_n) = \sqrt{\frac{\widehat{p}_n(1-\widehat{p}_n)}{n}}.$$

- In addition we know that the MLE for τ is $\widehat{\tau}_n = h(\widehat{p}_n)$ and

$$h'(p) = \frac{\partial}{\partial p}\big[\log(p) - \log(1-p)\big] = \frac{1}{p} + \frac{1}{1-p} = \frac{1-p+p}{p(1-p)} = \frac{1}{p(1-p)}.$$

- Hence, according to the δ–method,

$$\widehat{\mathrm{se}}(\widehat{\tau}_n) = \left|h'(\widehat{p}_n)\right|\widehat{\mathrm{se}}(\widehat{p}_n) = \frac{1}{\sqrt{n\,\widehat{p}_n(1-\widehat{p}_n)}}$$

Normal Confidence Intervals

The δ–method / General multiparametric models (I)

- These ideas can directly be extended to models with several parameters

$$\mathcal{F} = \left\{ f(x \mid \boldsymbol{\theta}) \,:\, \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^k \right\} \qquad (\text{typically } k << n)$$

- Let $\widehat{\boldsymbol{\theta}}$ be the MLE and $\ell_n(\boldsymbol{\theta}) = \sum_i \log f(X_i \mid \boldsymbol{\theta})$ the log–likelihood function.
- To build the Fisher Information Matrix we first need the $(k \times k)$ Hessian $\mathbb{H}$

$$\mathbb{H}_{r,r} = \frac{\partial^2}{\partial \theta_r^2} \ell_n(\boldsymbol{\theta}) \quad \text{and} \quad \mathbb{H}_{r,s} = \frac{\partial^2}{\partial \theta_r \partial \theta_s} \ell_n(\boldsymbol{\theta})$$

$$\mathcal{I}_n(\boldsymbol{\theta}) = - \begin{bmatrix} \mathbb{E}_{\boldsymbol{\theta}}\left(\mathbb{H}_{1,1}\right) & \mathbb{E}_{\boldsymbol{\theta}}\left(\mathbb{H}_{1,2}\right) & \cdots & \mathbb{E}_{\boldsymbol{\theta}}\left(\mathbb{H}_{1,k}\right) \\ \mathbb{E}_{\boldsymbol{\theta}}\left(\mathbb{H}_{2,1}\right) & \mathbb{E}_{\boldsymbol{\theta}}\left(\mathbb{H}_{2,2}\right) & \cdots & \mathbb{E}_{\boldsymbol{\theta}}\left(\mathbb{H}_{2,k}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{E}_{\boldsymbol{\theta}}\left(\mathbb{H}_{k,1}\right) & \mathbb{E}_{\boldsymbol{\theta}}\left(\mathbb{H}_{k,2}\right) & \cdots & \mathbb{E}_{\boldsymbol{\theta}}\left(\mathbb{H}_{k,k}\right) \end{bmatrix} \quad \rightsquigarrow \quad \mathbb{V} = \mathbb{V}_n(\boldsymbol{\theta}) = \mathcal{I}_n^{-1}(\boldsymbol{\theta}).$$

Theorem

- Under appropriate regularity conditions, $\left(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right) \dot{\sim} \mathsf{N}_k\left(0, \mathbb{V}\right)$.
- Also, if $\widehat{\theta}_r$ is the r^{th} element of $\widehat{\boldsymbol{\theta}}$, then $\frac{\widehat{\theta}_r - \theta_r}{\widehat{\text{se}}_r} \xrightarrow{d} \mathsf{N}_1(0,1)$, with $\widehat{\text{se}}_r^2 = \mathbb{V}_{r,r}$.
- The approximate covariance of $\widehat{\theta}_r$ and $\widehat{\theta}_s$ is $\mathbb{V}_{r,s}$.

Normal Confidence Intervals

The δ–method / General multiparametric models (II)

- Of course there is also a multiparameter δ–method (MANY-TO-ONE map)
- Let $\tau = h(\boldsymbol{\theta})$ and let $\nabla h = \left[\frac{\partial h}{\partial \theta_1} \cdots \frac{\partial h}{\partial \theta_k}\right]^{\mathrm{T}}$ be the gradient of $h : \mathbb{R}^k \mapsto \mathbb{R}$.

Theorem / Multiparameter δ–method

Suppose that $\widehat{\nabla} h = \nabla h\big(\widehat{\boldsymbol{\theta}}\big) \neq 0$ and let $\widehat{\tau} = h\big(\widehat{\boldsymbol{\theta}}\big)$ and $\widehat{\mathbb{V}} = \mathbb{V}\big(\widehat{\boldsymbol{\theta}}\big)$. Then

$$\frac{\widehat{\tau} - \tau}{\widehat{\mathrm{se}}\big(\widehat{\tau}\big)} \xrightarrow{d} \mathrm{N}_1\big(0, 1\big) \quad \text{where} \quad \widehat{\mathrm{se}}\big(\widehat{\tau}\big) = \sqrt{\big(\widehat{\nabla} h\big)^{\mathrm{T}}\, \widehat{\mathbb{V}}\, \big(\widehat{\nabla} h\big)}.$$

Example (numerical shortcut via Newton–like optimization methods)

- Let $\{X_1, \ldots, X_n\} \overset{\text{IID}}{\sim} \mathrm{N}_1(\mu, \sigma^2)$, and $\tau = h(\mu, \sigma) = \frac{\sigma}{\mu}$, then

$$\nabla h(\mu, \sigma) = \left[\frac{\partial h}{\partial \mu} \; \frac{\partial h}{\partial \sigma}\right]^{\mathrm{T}} = \left[-\frac{\sigma}{\mu^2} \; \frac{1}{\mu}\right]^{\mathrm{T}}.$$

- After a bit of algebra, it can be show that

$$\mathcal{I}_n(\mu, \sigma) = \begin{bmatrix} \frac{n}{\sigma^2} & 0 \\ 0 & \frac{2n}{\sigma^2} \end{bmatrix} \quad \rightsquigarrow \quad \mathbb{V}(\mu, \sigma) = \mathcal{I}_n^{-1}(\mu, \sigma) = \frac{1}{n} \begin{bmatrix} \sigma^2 & 0 \\ 0 & \frac{\sigma^2}{2} \end{bmatrix}$$

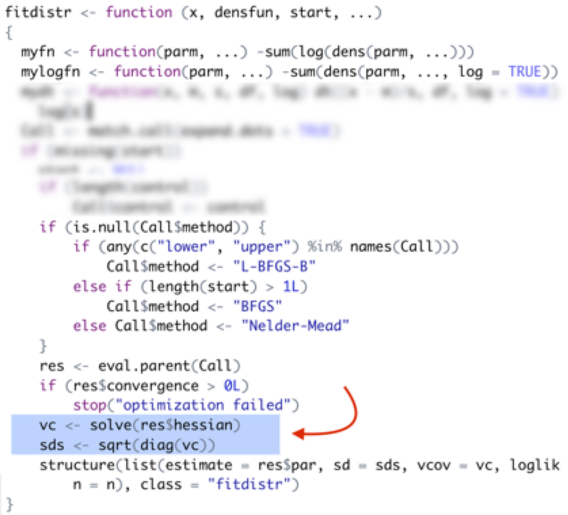
- Thus, in the end,

$$\widehat{\mathrm{se}}\big(\widehat{\tau}\big) = \frac{1}{\sqrt{n}} \sqrt{\frac{1}{\widehat{\mu}^4} + \frac{\widehat{\sigma}^2}{\widehat{\mu}^2}}.$$

Normal Confidence Intervals

The δ–method / General multiparametric models (III)

- Notice that all we need is an approximation to the (inverse of) the Hessian at the MLE.
- This quantity can be obtained (numerically) from any Newton–like 2nd order optimization method.
- As an example, take a quick look at the last few lines of the `fitdistr()` function from the `MASS` package.



```
fitdistr <- function (x, densfun, start, ...)
{
  myfn <- function(parm, ...) -sum(log(dens(parm, ...)))
  mylogfn <- function(parm, ...) -sum(dens(parm, ..., log = TRUE))
  [illegible]
    [illegible]
  [illegible]
  [illegible]
    [illegible]
    [illegible]
      [illegible]
    if (is.null(Call$method)) {
        if (any(c("lower", "upper") %in% names(Call)))
            Call$method <- "L-BFGS-B"
        else if (length(start) > 1L)
            Call$method <- "BFGS"
        else Call$method <- "Nelder-Mead"
    }
    res <- eval.parent(Call)
    if (res$convergence > 0L)
        stop("optimization failed")
    vc <- solve(res$hessian)
    sds <- sqrt(diag(vc))
    structure(list(estimate = res$par, sd = sds, vcov = vc, loglik
        n = n), class = "fitdistr")
}
```

Parametric Confidence Intervals (II)

The Parametric Bootstrap

Parametric Bootstrap

Generalities

The computational way...

- For parametric models, standard errors and confidence intervals may also be estimated using the parametric bootstrap.
- There is only one change:
  - In the nonparametric bootstrap, we sample $\{X_1^\star, \ldots, X_n^\star\}$ from the empirical distribution $\widehat{F}_n(\cdot)$.
  - In the parametric bootstrap we sample instead from $f\left(\cdot \,|\, \widehat{\theta}_n\right)$, where $\widehat{\theta}_n$ could be, for example the MLE or any other consistent estimator of the parameter θ.
- The bootstrap is much easier than the δ–method. On the other hand, the δ–method has the advantage that it gives a closed form, computationally "light" expression for the standard error.

The Boostraps

Analogies

The Nonparametric Bootstrap

	Real World	Bootstrap World
True Distribution	$F_X(\cdot)$	$\widehat{F}_n(\cdot)$
Sample/Data	$\{X_1, \dots, X_n\}$ IID $F_X(\cdot)$	$\{X_1^\star, \dots, X_n^\star\}$ IID $\widehat{F}_n(\cdot)$
ECDF	$\widehat{F}_n(\cdot)$	$F_n^\star(\cdot)$
Parameter	$\theta = g(F_X)$	$\widehat{\theta}_n = g(\widehat{F}_n)$
Estimator	$\widehat{\theta}_n = g(\widehat{F}_n)$	$\theta_n^\star = g(F_n^\star)$

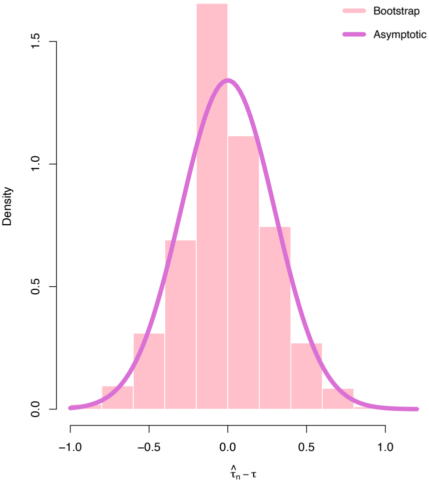
The Parametric Bootstrap

	Real World	Bootstrap World
Parameter	θ	$\widehat{\theta}_n$
True Distribution	$F_{\theta(\cdot)}$	$F_{\widehat{\theta}_n}(\cdot)$
Sample/Data	$\{X_1, \dots, X_n\}$ IID $F_\theta(\cdot)$	$\{X_1^\star, \dots, X_n^\star\}$ IID $F_{\widehat{\theta}_n}(\cdot)$
Estimator	$\widehat{\theta}_n = g(X_1, \dots, X_n)$	$\theta_n^\star = g(X_1^\star, \dots, X_n^\star)$

The Parametric Boostrap

Example

```r
# Get the weed data
load("meweed.RData")
n  = length(meweed)
tt = prop.table(table(meweed)); tt

# MLE + Asymptotic SE for "p"
p.hat = as.numeric(tt["yes"]); p.hat
se.p  = sqrt(p.hat*(1 - p.hat)/n)

# MLE + Asymptotic SE for "tau"
t.hat = log(p.hat/(1 - p.hat)); t.hat
se.t  = 1/sqrt(n * p.hat * (1 - p.hat) )

# Main loop
set.seed(123)
B = 1000
t.boot = rep(NA, B)
for (b in 1:B){
  x.boot = rbinom(n,     # sim a Ber(p.hat)
                 size = 1, prob = p.hat)
  p.star = mean(x.boot)
  t.boot[b] = log(p.star/(1 - p.star))
}
# Parametric bootstrap (vs) delta-Method
c(se.boot = sd(t.boot), se.delta = se.t)
# se.boot   se.delta
# 0.3028987 0.2974248
```



The Parametric Boostrap

Parametric vs Nonparametric

- The nonparametric bootstrap is nonparametric (surprise!) $\rightsquigarrow$ it always does the right thing, except when it doesn't...
- ...it doesn't work when the sample size is too small or when the data are not IID or when the parameter $\theta = g(F_X)$ is not a "nice" enough function of the true unknown distribution.
- The parametric bootstrap is parametric (surprise!) $\rightsquigarrow$ it is always wrong when the model is wrong (does not contain the true unknown distribution).
- On the other hand, when the parametric bootstrap does the right thing (when the statistical model is "almost" correct), it does a much better job at smaller sample sizes than the nonparametric bootstrap.

The Parametric Boostrap

Wrapping up

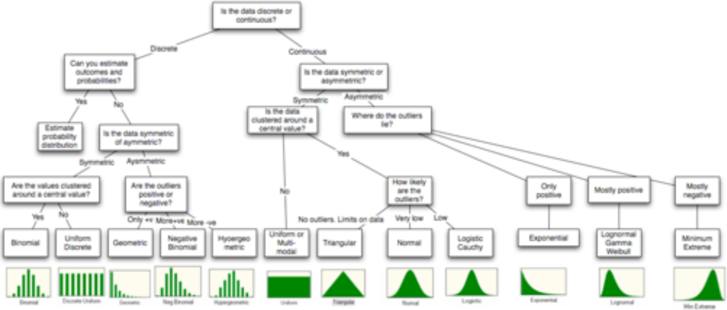
Achtung Achtung!

- When the parameter θ is defined in terms of the parametric statistical model and can only be estimated using the parametric model (by maximum likelihood perhaps), the statistical model needs to be correct for the parameter estimate $\widehat{\theta}_n$ to make sense.
- Since we already need the statistical model to be correct, the parametric bootstrap to get, for example, standard errors, is the logical choice.
- Simulation from the parametric model $F_{\widehat{\theta}_n}(\cdot)$ is **not** analogous to finite population sampling.
- Instead we simulate the parametric model: easy when R has a function to provide such random simulations!

Model Selection

Model Criticism & Selection

- How would we ever know that the distribution that generated the data belongs (or at least is close to) the parametric model we have chosen?
- One option is to informally check the assumptions encoded by our parametric model by inspecting plots of the data (e.g. histograms).
- There is even a decision tree available `in the wild` to drive your choice:



- A formal way to test a parametric model is to use a goodness-of-fit test. We will briefly talk about it soon.
- Here we will take another route...

Model Criticism & Selection

Big Heads–up

Model selection is a huge topic with many alternative perspectives.
Here we will just scratch the surface.

Example

- Suppose we have data $\{X_1, \ldots, X_n\}$ on times–to–failures for n HD's.
- You want to model the distribution $F_X(\cdot)$ of X.

Some popular models are:

1. $\mathcal{M}_1$: the exponential distribution $f_X(x|\lambda) = \lambda \mathrm{e}^{-\lambda \cdot x}$.
2. $\mathcal{M}_2$: the gamma distribution $f_X(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\, x^{\alpha-1}\, \mathrm{e}^{-\beta \cdot x}$.
3. $\mathcal{M}_3$: the log–normal distribution $\log X \sim \mathsf{N}(\mu, \sigma^2)$.

The most common model selections methods are:

1. Akaike Information Criterion (AIC) and related methods like Mallows' C_p.
2. Bayesian Information Criterion (BIC) and related methods like Minimum Description Length (MDL), Watanabe–Akaike Information Criterion (WAIC) and Bayesian model selection more in general.
3. Some variant of sample-splitting and cross-validation.

Model Criticism & Selection

Big Heads–up

Model selection is a huge topic with many alternative perspectives.
Here we will just scratch its (unsupervised) surface.

Example

- Suppose we have data $\{X_1, \ldots, X_n\}$ on times–to–failures for n HD's.
- You want to model the distribution $F_X(\cdot)$ of X.

Some popular models are:

1. $\mathcal{M}_1$: the exponential distribution $f_X(x|\lambda) = \lambda \mathrm{e}^{-\lambda \cdot x}$ (nested in the next).
2. $\mathcal{M}_2$: the gamma distribution $f_X(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\, x^{\alpha-1}\, \mathrm{e}^{-\beta \cdot x}$.
3. $\mathcal{M}_3$: the log–normal distribution $\log X \sim \mathrm{N}(\mu, \sigma^2)$.

Goals

We need to distinguish between two goals:

1. Find the model that gives the **best prediction**, without assuming that any of the models are correct ⇝ AIC and cross-validation.
2. Assume one of the models is the **true** model and...find the "true" model! ⇝ BIC.

⇝ Here we will focus on AIC leaving the BIC for our later Bayesian selves ⇜

Model Criticism & Selection

Akaike Information Criterion / Practice (I)

- Suppose we have a family of parametric models $\mathfrak{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_k\}$, where

$$\mathcal{M}_j = \{f_j(x|\boldsymbol{\theta}_j) : \boldsymbol{\theta}_j \in \Theta_j\}, \qquad \forall\, j \in \{1, \ldots, k\}.$$

- Suppose also that we have data $\{X_1, \ldots, X_n\}$ drawn from some density $f_X(\cdot)$.

WE DO NOT ASSUME THAT $f_X(\cdot)$ IS IN ANY OF THE MODELS IN $\mathfrak{M}$

- Let $\widehat{\boldsymbol{\theta}}_j$ be the MLE from model $j \in \{1, \ldots, k\}$, hence

$$\widehat{f}_j(x) \overset{\text{def.}}{=} f_j\big(x\big|\widehat{\boldsymbol{\theta}}_j\big) \text{ is an estimate of } f_X(\cdot) \text{ based on model } \mathcal{M}_j.$$

- To quantify how good $\widehat{f}_j(x)$ is as an estimate of $f_X(\cdot)$, as **loss function** we pick the Kullback–Leibler divergence (not a distance only 'cause it's not symmetric)

$$\mathrm{L}\big(f_X, \widehat{f}_j\big) = D_{\mathrm{KL}}\big(f_X \,\|\, \widehat{f}_j\big) = \int_{\mathcal{X}} f_X(x) \log \frac{f_X(x)}{\widehat{f}_j(x)}\, \mathrm{d}x$$

$$= \underbrace{\int_{\mathcal{X}} f_X(x) \log f_X(x)\, \mathrm{d}x}_{-H(f_X) \text{ entropy of } f_X(\cdot)} + \underbrace{\left(-\int_{\mathcal{X}} f_X(x) \log \widehat{f}_j(x)\, \mathrm{d}x\right)}_{H(f_X, \widehat{f}_j) \text{ cross–entropy}}$$

- The 1^{st} term does not depend on $j \Rightarrow \min_j \mathrm{L}\big(f_X, \widehat{f}_j\big) = \max_j -H\big(f_X, \widehat{f}_j\big)$

Model Criticism & Selection

Akaike Information Criterion / Practice (II)

$$\text{Select } \mathcal{M}_{\hat{j}} \text{ s.t. } \hat{j} = \underset{j\in\{1,\ldots,k\}}{\operatorname{argmin}} \int_{\mathcal{X}} f_X(x) \log \hat{f}_j(x)\, \mathrm{d}x = \underset{j\in\{1,\ldots,k\}}{\operatorname{argmin}} \mathbb{E}_{f_X}\Big(\log \hat{f}_j(X)\Big)$$

- The risk $R_j = \mathbb{E}_{f_X}\Big(\log \hat{f}_j(X)\Big)$ depends on the unknown $f_X(\cdot)$ ⇝ let's estimate it!
- Intuitively, we might think that the plug–in estimator is somewhat good

$$\tilde{R}_j = \frac{1}{n}\sum_{i=1}^{n} \log \hat{f}_j(X_i) = \frac{\ell_j(\hat{\boldsymbol{\theta}}_j)}{n} \overset{\texttt{Emp.Risk}}{\rightsquigarrow} \ell_j(\hat{\boldsymbol{\theta}}_j) \text{ is the log–likelihood for model } \mathcal{M}_j.$$

- However, this estimate is **very optimistic/downward biased** because the data are being used twice: 1^{st} to get the MLE and 2^{nd} to estimate the R_j.
- Akaike, a Japanese statistician who died recently (2009), showed that the bias is approximately d_j/n where $d_j = \dim(\Theta_j)$. Therefore, as predictive score, we use

$$\hat{R}_j = \frac{1}{n}\cdot\ell_j(\hat{\boldsymbol{\theta}}_j) - \frac{1}{n}\cdot d_j \quad \overset{\texttt{define}}{\rightsquigarrow} \quad \text{AIC}_j = 2\cdot n\cdot\hat{R}_j = \underbrace{2\cdot\ell_j(\hat{\boldsymbol{\theta}}_j) - 2\cdot d_j}_{\text{complexity penalized log-likelihood}}$$

- Clearly maximizing $\hat{R}_j$ is the same as maximizing AIC_j over j.
- Why do we multiply by $2n$? Just for historical reasons. We can multiply by any constant. In fact, different texts use different versions of AIC.

Model Criticism & Selection

Cross–Validation (I)

- Various flavors: in general, the data are split into a training set and a test set.
- The models are fit on the training set and are used to predict the test set.
- Usually, many such splits are used and the result are averaged over splits. Here, to keep things simple, we will use a single split.
- Suppose we have a family of parametric models $\mathfrak{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_k\}$, where

$$\mathcal{M}_j = \big\{ f_j(x|\boldsymbol{\theta}_j) : \boldsymbol{\theta}_j \in \Theta_j \big\}, \qquad \forall\, j \in \{1, \ldots, k\}.$$

- For simplicity, assume there are $2n$ data points and randomly split them in two halves denoted by $Tr_n = \{X_1, \ldots, X_n\}$ and $Te_n = \{X_1^\star, \ldots, X_n^\star\}$.
- Use Tr_n to find the MLE's $\{\widehat{\boldsymbol{\theta}}_1, \ldots, \widehat{\boldsymbol{\theta}}_k\}$ for each model in $\mathfrak{M}$.
- Now, based on Te_n define

$$\widehat{\mathrm{CV}}_j = \frac{1}{n} \sum_{i=1}^{n} \log f\left(X_i^\star \,\middle|\, \widehat{\boldsymbol{\theta}}_j\right).$$

- Note that, in this case, $\mathbb{E}\big(\widehat{\mathrm{CV}}_j\big) = R_j = \mathbb{E}_{f_X}\Big(\log \widehat{f}_j(X)\Big)$ directly since there is **no bias**: the estimator $\widehat{\boldsymbol{\theta}}_j$ is independent from Te_n.
- Our theoretical analysis can go even further...

Model Criticism & Selection

Cross–Validation (II)

- IF we assume $|\log f(x\,|\,\boldsymbol{\theta})| \leqslant B < \infty$, by Hoeffding + union bound we get

$$\mathbb{P}\left(\max_j \left|\widehat{\mathrm{CV}}_j - R_j\right| > \epsilon\right) \leqslant 2\cdot k\cdot \mathrm{e}^{-\frac{2n\epsilon^2}{2B^2}}.$$

- Equivalently, for a fixed $\alpha \in (0,1)$ and $\epsilon_n(\alpha) = \sqrt{\frac{2B^2\log(2k/\alpha)}{n}}$, then

$$\mathbb{P}\left(\max_j \left|\widehat{\mathrm{CV}}_j - R_j\right| < \epsilon_n(\alpha)\right) \geqslant 1-\alpha \qquad \heartsuit$$

- Hence, IF we choose $\widehat{j} = \text{argmax}_j \widehat{\mathrm{CV}}_j$, then, with probability at least $(1-\alpha)$,

$$R_{\widehat{j}} \;\overset{\heartsuit}{\leqslant}\; \widehat{\mathrm{CV}}_{\widehat{j}} + \epsilon_n(\alpha) \;\overset{j^\star = \mathrm{argmin}_j R_j}{\leqslant}\; \widehat{\mathrm{CV}}_{j^\star} + \epsilon_n(\alpha) \;\overset{\heartsuit}{\leqslant}\; R_{j^\star} + 2\cdot\epsilon_n(\alpha)$$

- With high probability, the model we select will be sub-optimal by at most $2\cdot\epsilon_n$.
- This argument can be improved and also applies, under a suitable loss–function, to regression, classification etc.

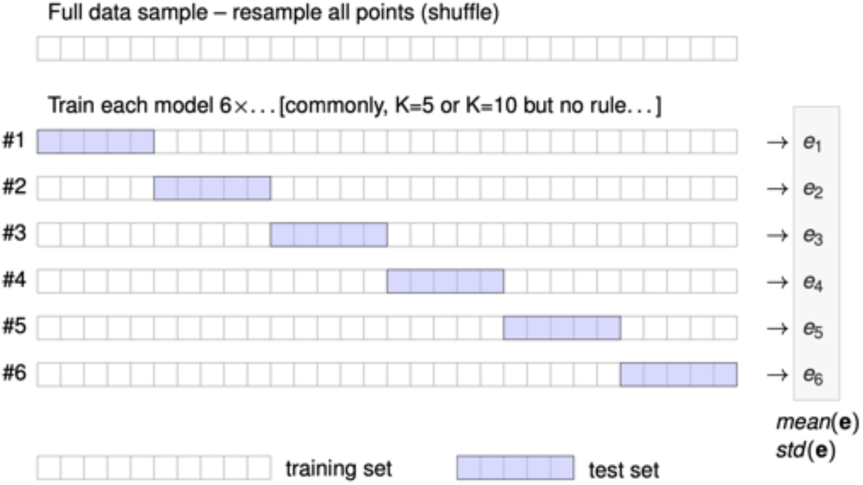
Please notice...

- We have made essentially no assumptions or approximations (the bound on $\log f$ can be relaxed).
- The beauty of cross-validation is its simplicity and generality.
- It can be shown that AIC and cross–validation have very similar behavior. But, cross-validation works under weaker conditions.

Model Criticism & Selection

Cross–Validation (III)

K–fold Cross–Validation



Full data sample – resample all points (shuffle)

Train each model 6×. . . [commonly, K=5 or K=10 but no rule. . .]

#1 → e_1

#2 → e_2

#3 → e_3

#4 → e_4

#5 → e_5

#6 → e_6

$mean(\mathbf{e})$
$std(\mathbf{e})$

training set

test set

Here e_k denotes the value of the loss/criterion function.

Model Criticism & Selection

BIC + Some final remarks

Bayesian Information Criterion in a Nutshell

- Practically, the BIC criterion is AIC but with a harsher constant/penalty:

$$\mathrm{BIC}_j = \ell_j(\widehat{\boldsymbol{\theta}}_j) - \left(\frac{\log n}{n}\right) \cdot d_j \quad \rightsquigarrow \quad \text{complexity penalized log-likelihood}$$

- Despite its name, this approach, proposed by Gideon Schwartz in 1978, is not information-theory driven but it comes from a Bayesian argument.
- More specifically, assume we place prior probabilities $\{\lambda_1, \ldots, \lambda_k\}$ over each model $\mathcal{M}_j$ + a prior distributions $p_j(\boldsymbol{\theta}_j)$ on its parameter. Then, by Bayes'rule:

$$\Pr(\mathcal{M}_j \,|\, \boldsymbol{X}_n) = \frac{\Pr(\boldsymbol{X}_n \,|\, \mathcal{M}_j)\,\Pr(\mathcal{M}_j)}{\sum_s \Pr(\boldsymbol{X}_n \,|\, \mathcal{M}_s)\,\Pr(\mathcal{M}_s)} = \frac{\lambda_J \cdot \int \mathcal{L}_j(\boldsymbol{\theta}_j)\, p_j(\boldsymbol{\theta}_j)\,\mathrm{d}\boldsymbol{\theta}_j}{\sum_s \lambda_s \cdot \int \mathcal{L}_s(\boldsymbol{\theta}_s)\, p_j(\boldsymbol{\theta}_s)\,\mathrm{d}\boldsymbol{\theta}_s}$$

- Schwartz showed that $\log \Pr(\mathcal{M}_j \,|\, \boldsymbol{X}_n) \approx \mathrm{BIC}_j$ in large samples.
- IF the true density f_X is in one of the model and $\mathcal{M}_{j^\star}$ is the smallest one that contains it, then it can be shown that $\widehat{j} \overset{\mathrm{P}}{\to} j^\star$ where $\widehat{j}$ is the BIC-choice.

- A difficult problem: how to account for model selection when doing inference.
- Simplest option with lots of data: keep an hold out set. After model selection, we use the hold out data which is not affected by the model selection process.
- Another issue is interpretability: getting good predictions is not the only goal.
- We might sacrifice a bit of prediction accuracy to have a more interpretable model. This is an area of active research.

Model Criticism & Selection

⇝ ONE LAST WEAPON IN OUR ARSENAL ⇜

Statistical Hypothesis Testing

[...Goodness of Fit testing, more specifically...]

Addendum (A)

Nonparametric Maximum Likelihood

On the optimality of the ECDF and beyond (I)

- As already said, the likelihood function is central to parametric inference but most of its nice statistical properties vanish going nonparametrics.
- Nevertheless, we can still squeeze something "nonparametrically" good out of it.

Definition: Nonparametric Likelihood Function

- Let $\boldsymbol{X}_n = \{X_1, \ldots, X_n\}$ be a random sample and $\boldsymbol{x}_n = \{x_1, \ldots, x_n\}$ a realization.
- Let also $\mathcal{F}$ be the set of all CDF $F(\cdot)$ on $\mathbb{R}$, and define $F(x^-) = \lim_{t \uparrow x} F(t)$
- Then we call nonparametric likelihood function for the data $\boldsymbol{x}_n$, the functional $\mathsf{L} : \mathbb{R}^n \times \mathcal{F} \mapsto [0, 1]$, that maps any data–model pair $(\boldsymbol{x}_n, F)$ into

$$(\boldsymbol{x}_n, F) \mapsto \mathsf{L}(\boldsymbol{x}_n, F) = \prod_{i=1}^{n} \left(F(x_i) - F(x_i^-)\right) = \prod_{i=1}^{n} \mathbb{P}_F(\{x_i\}).$$

  where $\mathbb{P}_F(\cdot)$ denotes the probability measure induced by $F(\cdot)$.
- Please notice that here $F(\cdot)$ is an argument of $\mathsf{L}(\cdot, \cdot) \rightsquigarrow$ *functional*.

Theorem: Nonparametric MLE

- For every data point $\boldsymbol{x}_n$, the empirical CDF $\widehat{F}_n(\cdot)$ maximizes $\mathsf{L}(\boldsymbol{x}_n, \cdot)$ over $\mathcal{F}$.
- For this reason we say that the ECDF $\widehat{F}_n(\cdot)$ is the nonparametric maximum likelihood estimator of the unknown, underlying CDF $F(\cdot)$.

Nonparametric Maximum Likelihood

On the optimality of the ECDF and beyond (II)

Theorem: Nonparametric MLE (proof)

- Let $F_0(\cdot)$ be a generic candidate solution to the optimization problem:

$$\underset{F \in \mathcal{F}}{\operatorname{argmax}}\, L(\boldsymbol{x}_n, F).$$

- It is immediately clear that, to be the optimal solution, the measure induced $\mathbb{P}_{F_0}(\cdot)$ has to be discrete placing its entire mass on the observations $\{x_1, \dots, x_n\}$.
- Hence, any candidate NMLE is characterized by an n–tuple $\{w_i\}_{i=1}^n$ where $w_i = \mathbb{P}_{F_0}\big(\{x_i\}\big)$, so: $w_i \geqslant 0$ for any i, and $\sum_i w_i = 1$.
- Although $\mathcal{F}$ is an infinite–dimensional function space, this shows that the optimization of $L(\boldsymbol{x}_n, \cdot)$ over $\mathcal{F}$ is actually equivalent to a finite-dimensional optimization problem over the unit simplex in $\mathbb{R}^n$.
- Hence, we left to solve the following constrained optimization problem

$$\max_{\{w_1, \dots, w_n\}} \prod_{i=1}^n w_i \quad \text{subject to} \quad \sum_i w_i = 1 \;\text{ and }\; w_i \geqslant 0 \;\; \forall\, i \in \{1, \dots, n\}.$$

- The solution can easily be obtained by using the Lagrange multiplier method, and, as anticipated, it is given by $w_i = \frac{1}{n}$ for all $i \in \{1, \dots, n\}$.

□

⇝ Looking for something smoother? Penalize! ⇜

Addendum (B)

Model Criticism & Selection

Akaike Information Criterion / Theory (I)

Goal

- Focusing on one model by dropping the subscript j, we want to estimate:

$$R = \mathbb{E}_{f_X}\Big(\log \widehat{f}(X)\Big) = \int_{\mathcal{X}} f_X(x) \log f\big(x \,|\, \underbrace{\widehat{\boldsymbol{\theta}}}_{\text{MLE}}\big)\,\mathrm{d}x \quad \text{with} \quad d = \dim(\Theta).$$

- Our goal then, is to show that the bias–corrected log–likelihood satisfies

$$\widetilde{R} - \frac{d}{n} \approx R \quad \text{where} \quad \widetilde{R} = \frac{1}{n}\sum_{i=1}^{n} \log \widehat{f}(X_i) = \frac{1}{n}\ell(\widehat{\boldsymbol{\theta}}).$$

Notation & Background

1. Let $f(\cdot \,|\, \boldsymbol{\theta}_0)$ be the closest density (in the KL–sense) to the true model $f_X(\cdot)$.
2. Let $\ell(\boldsymbol{\theta}) = \log f(x \,|\, \boldsymbol{\theta})$, $\boldsymbol{s}(x \,|\, \boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} \log f(x \,|\, \boldsymbol{\theta})$ be the score function, $\mathbb{H}(x \,|\, \boldsymbol{\theta})$ be the matrix of second derivatives, and $\boldsymbol{S}_n = \frac{1}{n}\sum_i \boldsymbol{s}(X_i \,|\, \boldsymbol{\theta}_0)$.
3. Since we are in a misspecified model setup, the MLE $\widehat{\boldsymbol{\theta}}$ is still asymptotically Normal and unbiased but with an "adjusted" variance–covariance matrix, that is

$$\sqrt{n}\big(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\big) \overset{\text{def.}}{=} \boldsymbol{Z}_n \xrightarrow{d} \mathrm{N}_d\big(0, \mathbb{J}^{-1}\mathbb{V}\,\mathbb{J}^{-1}\big) \quad + \quad \sqrt{n}\cdot\boldsymbol{S}_n \xrightarrow{\text{CLT}} \mathrm{N}_d(0, \mathbb{V}),$$

where $\mathbb{J} = -\mathbb{E}_{f_X}\big(\mathbb{H}(X \,|\, \boldsymbol{\theta}_0)\big)$ and $\mathbb{V} = \mathbb{V}\mathrm{ar}_{f_X}\big(\boldsymbol{s}(X \,|\, \boldsymbol{\theta}_0)\big)$, so $\mathbb{J}\cdot\boldsymbol{Z}_n \approx \sqrt{n}\cdot\boldsymbol{S}_n$.

4. Let $\boldsymbol{\epsilon}$ be a random vector with mean $\boldsymbol{\mu}$ and covariance Σ, and $Q = \boldsymbol{\epsilon}^{\mathrm{T}}\mathbb{A}\,\boldsymbol{\epsilon}$, then

$$\mathbb{E}(Q) = \mathrm{trace}\big(\mathbb{A}\,\Sigma\big) + \boldsymbol{\mu}^{\mathrm{T}}\mathbb{A}\,\boldsymbol{\mu} \qquad \text{(quadratic form)}$$

Model Criticism & Selection

Akaike Information Criterion / Theory (II)

Goal

$$\widetilde{R} - \frac{d}{n} \approx R = \int_{\mathcal{X}} f_X(x) \log f\big(x \big| \underbrace{\widehat{\boldsymbol{\theta}}}_{\text{MLE}}\big) \, \mathrm{d}x \quad \text{where} \quad \widetilde{R} = \frac{1}{n} \sum_{i=1}^{n} \log f\big(X_i \,|\, \widehat{\boldsymbol{\theta}}\big) = \frac{1}{n} \ell_n\big(\widehat{\boldsymbol{\theta}}\big)$$

Step 1: Taylor expand R around $\boldsymbol{\theta}_0$

$$R \approx \int_{\mathcal{X}} f_X(x) \Big(\log f(x \,|\, \boldsymbol{\theta}_0) + \big(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\big)^{\mathrm{T}} \boldsymbol{s}(x \,|\, \boldsymbol{\theta}_0) + \tfrac{1}{2} \big(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\big)^{\mathrm{T}} \mathbb{H}(x \,|\, \boldsymbol{\theta}_0) \big(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\big) \Big) \mathrm{d}x$$

$$= \int_{\mathcal{X}} f_X(x) \log f(x \,|\, \boldsymbol{\theta}_0) \mathrm{d}x - \tfrac{1}{2n} \boldsymbol{Z}_n^{\mathrm{T}} \mathbb{J} \, \boldsymbol{Z}_n = R_0 - \tfrac{1}{2n} \boldsymbol{Z}_n^{\mathrm{T}} \mathbb{J} \, \boldsymbol{Z}_n.$$

The 2nd term dropped out because, like the score function, it has mean 0.

Step 2: Taylor expand $\widetilde{R}$ around $\boldsymbol{\theta}_0$

$$\widetilde{R} \approx \tfrac{1}{n} \sum_{i=1}^{n} \Big(\log f(X_i \,|\, \boldsymbol{\theta}_0) + \big(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\big)^{\mathrm{T}} \boldsymbol{s}(X_i \,|\, \boldsymbol{\theta}_0) + \tfrac{1}{2} \big(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\big)^{\mathrm{T}} \mathbb{H}(X_i \,|\, \boldsymbol{\theta}_0) \big(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\big) \Big)$$

$$= R_0 + A_n + \big(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\big)^{\mathrm{T}} \boldsymbol{S}_n - \tfrac{1}{2n} \boldsymbol{Z}_n^{\mathrm{T}} \widetilde{\mathbb{J}}_n \, \boldsymbol{Z}_n \overset{(\heartsuit)}{\approx} R_0 + A_n + \tfrac{1}{\sqrt{n}} \boldsymbol{Z}_n^{\mathrm{T}} \boldsymbol{S}_n - \tfrac{1}{2n} \boldsymbol{Z}_n^{\mathrm{T}} \mathbb{J} \, \boldsymbol{Z}_n,$$

where $A_n = \frac{1}{n} \sum_i \big(\log f(X_i \,|\, \boldsymbol{\theta}_0) - R_0 \big)$ and $\widetilde{\mathbb{J}}_n = -\frac{1}{n} \sum_i \mathbb{H}(X_i \,|\, \boldsymbol{\theta}_0) \xrightarrow{P} -\mathbb{J}$.

Model Criticism & Selection

Akaike Information Criterion / Theory (III)

Goal

$$\widetilde{R} - \frac{d}{n} \approx R = \int_{\mathcal{X}} f_X(x) \log f\big(x \,|\, \underbrace{\widehat{\boldsymbol{\theta}}}_{\text{MLE}}\big)\, \mathrm{d}x \quad \text{where} \quad \widetilde{R} = \frac{1}{n}\sum_{i=1}^{n} \log f\big(X_i \,|\, \widehat{\boldsymbol{\theta}}\big) = \frac{1}{n}\ell_n\big(\widehat{\boldsymbol{\theta}}\big)$$

Up to now...

$$\big(\widetilde{R} - R\big) \approx A_n + \frac{\boldsymbol{Z}_n^{\mathrm{T}}\big(\sqrt{n}\,\boldsymbol{S}_n\big)}{n} \approx A_n + \frac{\boldsymbol{Z}_n^{\mathrm{T}}\mathbb{J}\,\boldsymbol{Z}_n}{n} \quad \text{with} \quad A_n = \frac{1}{n}\sum_i \big(\log f(X_i \,|\, \boldsymbol{\theta}_0) - R_0\big)$$

- We conclude that

$$\mathbb{E}\big(\widetilde{R} - R\big) \approx \mathbb{E}\big(A_n\big) + \mathbb{E}\left(\frac{\boldsymbol{Z}_n^{\mathrm{T}}\mathbb{J}\,\boldsymbol{Z}_n}{n}\right) = 0 + \frac{\operatorname{trace}\big(\mathbb{J}\,\mathbb{J}^{-1}\mathbb{V}\,\mathbb{J}^{-1}\big)}{n} = \frac{\operatorname{trace}\big(\mathbb{J}^{-1}\mathbb{V}\big)}{n}.$$

- Hence

$$R \approx \widetilde{R} - \frac{\operatorname{trace}\big(\mathbb{J}^{-1}\mathbb{V}\big)}{n}.$$

- IF the model is correct, then $\mathbb{J}^{-1} = \mathbb{V}$ so that $\operatorname{trace}\big(\mathbb{J}^{-1}\mathbb{V}\big) = \operatorname{trace}\big(\mathbb{I}_d\big) = d$.
- It is apparent that many approximations and assumptions are used, so AIC is just a very crude tool ⇝ cross–validation is much more reliable.