Maximum Likelihood Estimation

Wednesday, Course Week 3

Gherardo Varando

Exam

30 hours take home exam

- Start: 15 January 2019 at 09:00
- Deadline: 16 January 2019 at 15:00
- You can submit just a pdf (it can be generated with Rmarkdown)

The maximum likelihood

$$\mathcal{L}_n(\theta) = \prod_{i=1}^{n} f(X_i|\theta)$$

$$\ell_n(\theta) = \log\left(\mathcal{L}_n(\theta)\right)$$

MLE

The maximum likelihood estimator (MLE), is $\hat{\theta}$ the value of the parameter θ that maximizes $\mathcal{L}_n(\theta)$

Exponential distribution

$$\ell_n(\lambda) = n \log(\lambda) - \lambda \sum_{i=1}^{n} X_i$$

To find the maximum of the log-likelihood we compute the derivatives (first and second) of the log-likelihood.

$$\ell_n'(\lambda) = \frac{d\ell_n(\lambda)}{d\lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} X_i$$

$$\ell_n''(\lambda) = -\frac{n}{\lambda^2}$$

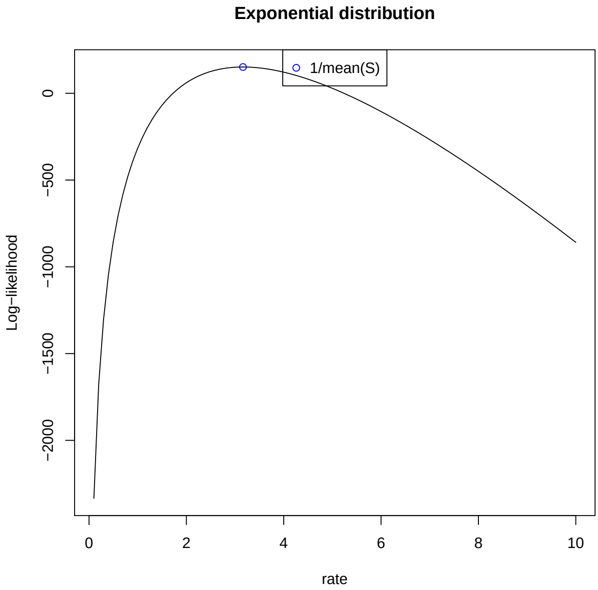
Thus we find the critical points solving the equation $\ell_n'(\lambda) = 0$

$$\frac{n}{\lambda} - \sum_{i=1}^{n} X_i = 0$$

$$\lambda = \frac{n}{\sum_{i=1}^{n} X_i} = \frac{1}{\overline{X}}$$

We have just one critical points and since $\ell_n''(\lambda) < 0$ the critical point is a maximum of the function.

$$\hat{\lambda} = \frac{1}{\overline{X}} \tag{MLE}$$


Exponential distribution
1/mean(S)
Log−likelihood
0
−500
−1000
−1500
−2000
0
2
4
6
8
10
rate

Uniform distribution

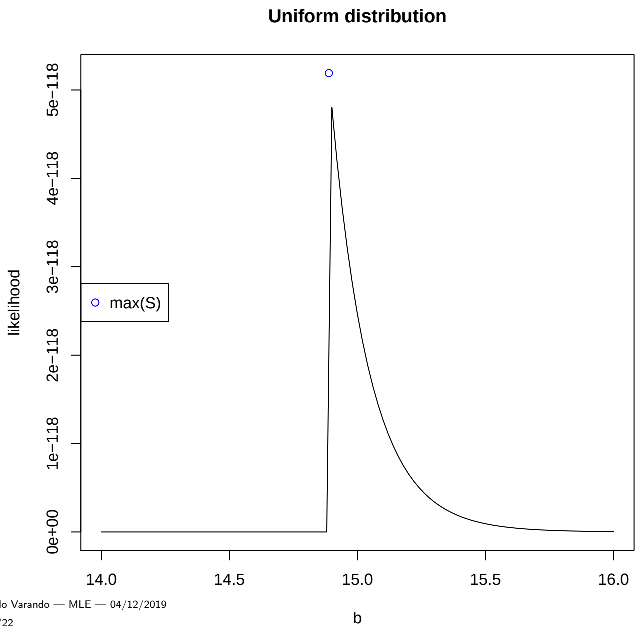
$$X_1, \ldots, X_n \sim Uniform([0, b]) \quad b > 0$$

$$f(x|b) = \begin{cases} 0 & x < 0 \\ \frac{1}{b} & 0 \leq x \leq b \\ 0 & x > b \end{cases}$$

$$\mathcal{L}_n(b) = \prod_{i=1}^{n} f(x|b) = \begin{cases} 0 & \text{if } \max\{X_1, \ldots, X_n\} > b \\ \frac{1}{b^n} & \text{if } \max\{X_1, \ldots, X_n\} \leq b \end{cases}$$

The maximum with respect to b is attained at the point

$$\hat{b} = \max\{X_1, \ldots, X_n\} \qquad \text{(MLE)}$$


Uniform distribution
likelihood
0e+00
1e−118
2e−118
3e−118
4e−118
5e−118
max(S)
14.0
14.5
15.0
15.5
16.0
b

Multiple parameters

If the parameter for our statistical model is multi-dimensional,

$$\theta = (\theta_1, \theta_2, \ldots, \theta_k)$$

If the log-likelihood has only one maximum, finding the MLE amounts to solving the system of k equations,

$$\nabla \ell_n(\theta) = \begin{pmatrix} \frac{\partial \ell_n(\theta)}{\partial \theta_1} \\ \frac{\partial \ell_n(\theta)}{\partial \theta_2} \\ \vdots \\ \frac{\partial \ell_n(\theta)}{\partial \theta_k} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$\nabla \ell_n(\theta)$ is called the **gradient** of $\ell_n(\theta)$ and geometrically represents the direction of maximum growth of the function $\ell_n(\theta)$

Example Gaussian distribution

$$X_1, \ldots, X_n \sim N(\mu, \sigma^2)$$

$$\mathcal{L}_n(\mu, \sigma) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(X_i - \mu)^2}{2\sigma^2}\right)$$

$$\ell_n(\mu, \sigma) = \sum_{i=1}^{n} - \log\left(\sigma\right) - \log\left(\sqrt{2\pi}\right) - \frac{(X_i - \mu)^2}{2\sigma^2}$$

$$\ell_n(\mu, \sigma) = -n\log(\sigma) - n\log(\sqrt{2\pi}) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(X_i - \mu)^2$$

$$\ell_n(\mu, \sigma) = -n \log(\sigma) - n \log(\sqrt{2\pi}) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (X_i - \mu)^2$$

We have to compute the gradient, that is the two partial derivatives with respect to μ and σ, and set them to 0.

$$\frac{\partial \ell_n(\mu, \sigma)}{\partial \mu} = \frac{1}{\sigma^2} \sum_{i=1}^{n} (X_i - \mu) = 0 \tag{1}$$

$$\frac{\partial \ell_n(\mu, \sigma)}{\partial \sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma^3} \sum_{i=1}^{n} (X_i - \mu)^2 = 0 \tag{2}$$

From equation (1) we easily obtain that

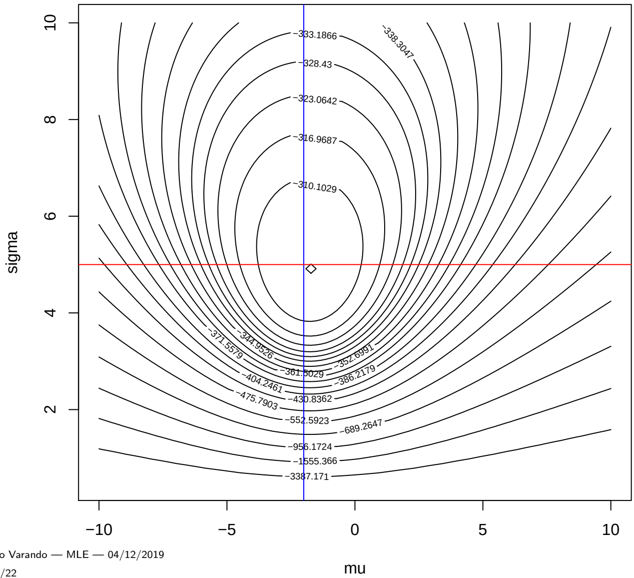
$$\hat{\mu} = \frac{\sum_{i=1}^{n} X_i}{n} = \overline{X}$$

The substituting $\hat{\mu}$ in equation (2) and solving for σ we obtain

$$\hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{\mu})^2}$$

We should now check that the value $(\hat{\mu}, \hat{\sigma})$ is actually a point of maximum for the log-likelihood. This can be done computing the Hessian matrix (that is second order derivatives) and check that is negative definite.

Contour plot of log likelihood for N(−2,25)



Finding the MLE

- We have seen examples where the MLE can be found analytically.
- However for a general situation our model can includes complicated distributions and dependencies among data.
- The likelihood function can easily became too complicated to be maximized analytically and thus we need **numerical methods**.

Optimization

Optimization is the task to find the maximum or minimum of a given function (of one or multiple variables), possibly with some **restrictions** on the variable space

- Maximizing f is equivalent to minimize $-f$
- Since usually optimization task are stated as minimization problems, we will re-state the MLE estimation problem as finding the **minimum of the minus log-likelihood**

$$-\ell_n(\theta) = -\sum_{i=1}^{n} \log\left(f(X_i|\theta)\right)$$

Iterative optimization algorithms

1. Start from an initial guess θ_0
2. Move to a new point θ_1
3. Check some stopping criteria, if not fulfilled repeat from (2)
4. Return last computed point θ_m

Iterative optimization algorithm can be achieved with different methods. Mainly we can identify:

- Derivative-free methods, do not use the derivative of the objective function
- Derivative-based methods, use information provided by the derivatives

In R we can use the function `optimize` and `optim` to perform optimization.

- `optimize` solves only one-dimensional optimization problems (in our case when we only have one parameter). It uses a derivative-free method that works with continuous functions.
- `optim` performs general multi dimensional optimization, using different methods:
    - `Nelder-Mead`, a derivative-free method, works relatively well for non-differentiable functions but it is relatively slow.
    - `BFGS` is a quasi-newton method that use only first derivatives (gradient)
    - `CG` conjugate gradient method, useful fot very large optimization problems
    - `SANN` use simulated annealing

Gradient descent

We see now a simple algorithm for optimization, more complicated methods are usually a refinement of this.

Gradient descent

Since the gradient $\nabla f(x)$ of a function f, indicates the direction of maximum growth, we can think of following the gradient (ascent) to find the maximum or follow the opposite of the gradient (descent) to find the minimum

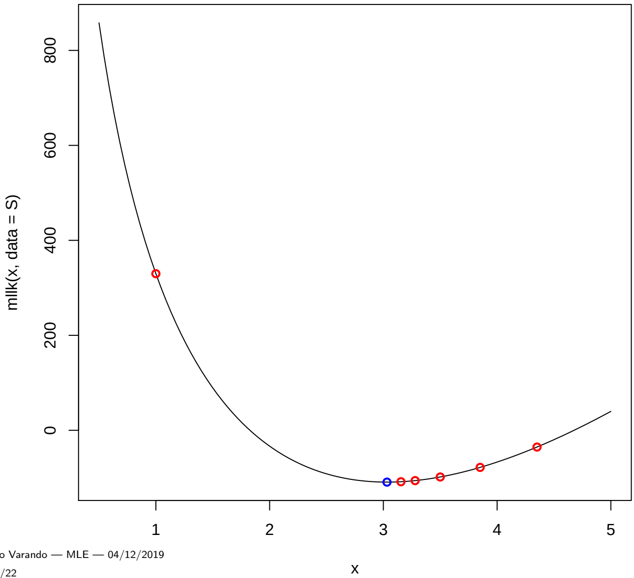
- Imagine the optimization problem as the problem of finding the highest point of a geographical map
- We can imagine to move in the landscape and at every moment we make a step in the steepest direction.

One dimensional optimization

Algorithm 1 One-dimensional gradient descent

Require: Objective function f, derivative f', initial point θ_0, maximum number of iterations M and tolerance ϵ

for $i = 1$ to M **do**
    $\theta_i = \theta_{i-1} - \lambda f'(\theta_i)$
    if $\frac{|f(\theta_i) - f(\theta_{i-1})|}{|f(\theta_{i-1})| + \epsilon} < \epsilon$ **then**
        end loop
    end if
end for
return θ_i



mllk(x, data = S)
0
200
400
600
800
1
2
3
4
5
x

Multi-dimensional optimization

Algorithm 2 Multi-dimensional gradient descent

Require: Objective function f, gradient ∇f, initial point θ_0, maximum number of iterations M and tolerance ϵ

```
for i = 1 to M do
    θ_i = θ_{i-1} − λ∇f(θ_i)
    if |f(θ_i) − f(θ_{i-1})| / (|f(θ_{i-1})| + ε) < ε then
        end loop
    end if
end for
return θ_i
```

for $i = 1$ to M **do**

  $\theta_i = \theta_{i-1} - \lambda \nabla f(\theta_i)$

  if $\frac{|f(\theta_i) - f(\theta_{i-1})|}{|f(\theta_{i-1})| + \epsilon} < \epsilon$ **then**

    end loop

  end if

end for

return θ_i

Contour plot of minus log likelihood for N(−2,25)