Recap

- **ML method to estimate θ:**

- Find the value of θ that maximizes the likelihood function, $L(\theta) = \prod_{i=1}^{n} \underbrace{f_\theta(x_i)}_{\text{PDF/PMF of } X_i}$, with respect to θ. This value is $\hat{\theta}_{MLE}$.

- generally Easier to maximize the log-likelihood function, $\log L(\theta) = \sum_{i=1}^{n} \log f_\theta(x_i)$.

- Either direct maximization or differentiation technique or numerical maximization

- Remember the pitfalls in numerical maximization.

Finding standard error (SE) of $\hat{\theta}$

Need to report $\hat{SE}(\hat{\theta})$, along with an estimate.

Simple cases:

$$\left.\begin{array}{ll} ① & \hat{SE}[\bar{X}] = \dfrac{s}{\sqrt{n}}, \quad s = \text{SD of sample.} \\ ② & \hat{SE}[\hat{p}] = \sqrt{\dfrac{\hat{p}(1-\hat{p})}{n}} \end{array}\right\} \text{see before}$$

③ In general, it is difficult to find closed-form expression for SE's.

Two general techniques:

① Bootstrap (Later)

simulation-based

② Large sample properties of MLE.

— hold only when n is large.

Large sample properties of MLE $\hat{\theta}$ of θ

domain of the dist. | set of all possible values of X.

Result: Assume that $\{x : f_\theta(x) > 0\}$ is free of θ. Then, under certain conditions when n is large,

Hessian matrix ~~and~~ (or observed Fisher information matrix)

$$\hat{\theta} \approx N(\theta, \hat{I}^{-1}), \text{ where } \hat{I} = -\left.\frac{\partial^2 \cancel{L(\theta)}\,\log L(\theta)}{\partial \theta^2}\right|_{\theta=\hat{\theta}}.$$

Case 1: θ is scalar. Then, $\widehat{SE}(\hat{\theta}) \cancel{=}\approx \sqrt{\hat{I}^{-1}}$

Case 2: θ is a vector, say, $\theta = (\theta_1, \ldots, \theta_d)$. In this case, $\hat{I}$ is a $d \times d$ matrix. Here $\widehat{SE}(\hat{\theta}_j) \cancel{=}\approx (j\text{-th diagonal element of } \hat{I}^{-1})^{1/2}$.

Properties:

Assumes that the fitted model is correct.

- Consistent
- (approx). unbiased if n is large
- Asymptotically normal
- Optimal if the assumed model holds
- Not a good choice if the assumed model does not hold

Recall: ① $X \sim \text{Uniform}(\theta)$

$$f_\theta(x) = \begin{cases} \frac{1}{\theta}, & 0 \le x \le \theta \\ 0, & \text{o/w.} \end{cases}$$

Domain $= [0, \theta]$

depends on θ.

② $X \sim \text{Exp}(\lambda)$.

$$f_\lambda(x) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0 \\ 0, & \text{o/w.} \end{cases}$$

Domain $= [0, \infty)$

free of θ.

③ $X \sim \text{Bernoulli}(p)$.

p

$$f_p(x) = p^x (1-p)^{1-x}, \quad x = 0, 1$$

Domain $= \{0, 1\} \longrightarrow$ free of p.

Using R to get MLE

Ex: Recall the CPU data — CPU times for $n = 30$ randomly chosen jobs (in seconds): 70, 36, 43, 69, 82, 48, 34, 62, 35, 15, 59, 139, 46, 37, 42, 30, 55, 56, 36, 82, 38, 89, 54, 25, 35, 24, 22, 9, 56, 19. Graphics suggested that the distribution of these CPU times may be right-skewed. Suppose we ***assume*** that the parent distribution is Gamma (α, λ), with both parameters unknown. What are MLE's of these parameters?

Recall: $f(x) = \frac{\lambda^\alpha}{\Gamma\alpha} x^{\alpha-1} e^{-\lambda x}, \quad x > 0, \quad \alpha > 0, \lambda > 0$

positive

See book: For MOME.

MLE: $\log L(\theta) = \sum_{i=1}^{n} \log \left[\frac{\lambda^\alpha}{\Gamma\alpha} x_i^{\alpha-1} e^{-\lambda x_i} \right]$.

(α, λ)

— Difficult to maximize in closed form.

```
# We will continue working with the CPU data
# that we saw earlier

cpu <- scan(file="cputime.txt")

# Negative of log-likelihood function assuming gamma
# parent distribution
```

(α, λ) data

```
neg.loglik.fun <- function(par, dat)
{
result <- sum(dgamma(dat, shape=par[1], rate=par[2],
log=TRUE))
return(-result)
}
```

α λ

```
# Minimize -log (L), i.e., maximize log (L)
```

? optim

initial values are close to MOME see book.

```
ml.est <- optim(par=c(3, 0.1), fn=neg.loglik.fun,
```

minimizes by default.

```
method = "L-BFGS-B", lower=rep(0,2), hessian=TRUE,
dat=cpu)
```

allows constraints. $\alpha > 0,\ \lambda > 0$.

```
# > ml.est
# $par
# [1] 3.63149628 0.07529459
```

— $\hat{\alpha}, \hat{\lambda}$

```
# $value
# [1] 136.561
```

= min value of $\log L(\alpha,\lambda)$.

```
# $counts
# function gradient
     # 20        20

# $convergence
# [1] 0
```

— good.

```
# $message
```

```
# [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"

# $hessian
              # [,1]        [,2]
# [1,]    9.501374  -398.4584
# [2,] -398.458449 19223.5065
# >
```

$\hat{I}$

Know:
$\hat{I}^{-1}$ = ~~matrix~~ var-cov matrix.

cov2cor($\hat{I}^{-1}$).
|
correlation.

```
# MLE

# > ml.est$par
# [1] 3.63149628 0.07529459
# >

# their standard errors

# > sqrt(diag(solve(ml.est$hessian)))
# [1] 0.89720941 0.01994668
```

$\hat{I}^{-1}$

diagonal elements of $\hat{I}^{-1}$ = variances of elements of $\hat{\theta}$.

$\hat{I}^{-1}_{ij}$ = element in ith row and jth col of $\hat{I}^{-1}$ = $\widehat{cov}(\hat{\theta}_i, \hat{\theta}_j)$

<u>Recall:</u> $\text{cov}(X,Y) = E[(X-E(X))(Y-E(Y))]$.

- hard to interpret

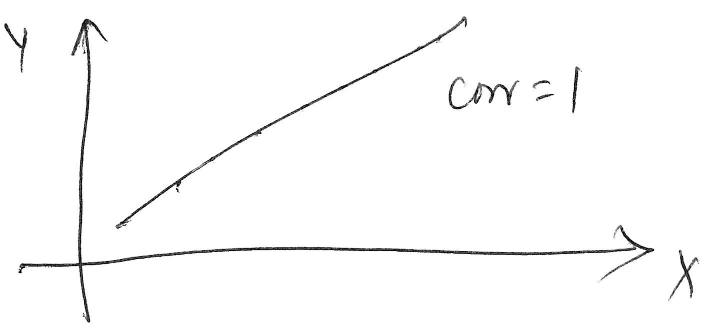
$$\text{corr}(X,Y) = \frac{\text{cov}(X,Y)}{SD(X)\cdot SD(Y)}$$

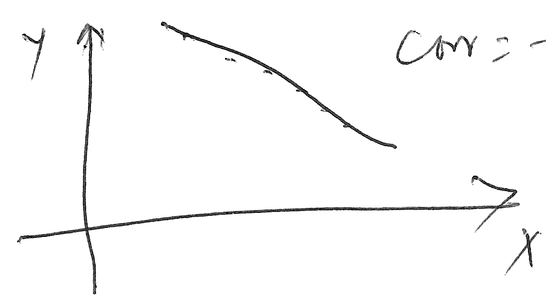
- $-1 \le \text{corr}(X,Y) \le 1$.

corr ≈ 0.

measure of strength of linear ~~linear~~ relationship.

In particular, $\text{corr}(X,Y) = \begin{cases} 1, & Y = aX+b,\ a>0 \\ -1, & a<0. \end{cases}$





Using R to get MOME

Ex: Suppose we have the following data on number of months elapsed before the next virus attack on a server:

$$0.49, 0.50, 0.11, 0.09, 0.05, 0.75, 0.34, 0.04, 0.26, 0.16.$$

We assume that the these data are i.i.d. observations from an Exponential (λ) distribution. What is the MOME of λ?

Know:

$$\frac{1}{\lambda} = E[X] = \bar{X}$$

$$E[X] = \int_{-\infty}^{\infty} x f(x)\, dx$$

$$\Rightarrow \quad \lambda = \frac{1}{\bar{X}} = \frac{1}{\hat{\lambda}_{MOME}}.$$

Here:

$$E[X] = \lambda \int_0^{\infty} x e^{-\lambda x}\, dx = \frac{1}{\lambda}$$

```
# Input the data

time.attack <- c(0.49, 0.50, 0.11, 0.09, 0.05,
0.75, 0.34, 0.04, 0.26, 0.16)

# Function to numerically compute mean
# of exponential distribution

exp.mean <- function(lambda){
fun <- function(x){x*dexp(x, rate=lambda)}
result <- integrate(fun, lower=0, upper=Inf)
return(result$value)}

# Equation whose solution is MOME

mome.eqn <- function(lambda, dat)
{exp.mean(lambda)-mean(dat)}
```

$E[X] - \bar{X}$

```
# >

# Already know that the answer should be:

# > 1/mean(time.attack)
# [1] 3.584229
# >
```

Note: It is much easier to get MOME directly rather than numerically.

```
# Solve this equation to get MOME

result <- uniroot(mome.eqn, lower=1, upper=5,
dat=time.attack)

# > result
# $root
# [1] 3.584233

# $f.root
# [1] -2.579698e-07

# $iter
# [1] 7

# $estim.prec
# [1] 6.103516e-05
```

$\gamma \quad E[X] - \bar{x} = 0.$

MOME

Large sample properties of MOME $\hat{\theta}$ of θ

- consistent
- Asymptotically normal
- $SE[\hat{\theta}]$ may be hard to find (will use bootstrap)
- optimality prop. of MLE doesn't hold.

MOME vs MLE

- Easy to compute
- No such property
-

- may be hard to compute
- optimality holds if n is large.