Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Δ.Π.Μ.Σ. Επιστήμης Δεδομένων
και Μηχανικής Μάθησης

Υπολογιστική Στατιστική και Στοχαστική Βελτιστοποίηση

Επιβλέπων: Δημήτρης Φουσκάκης
Καθηγητής Ε.Μ.Π.

June 29, 2021

Περιεχόμενα

1. **2**

1.(α')Monte Carlo Ολοκλήρωση 2

1.(β')Monte Carlo Εκτιμητής - Αμεροληψία - Τυπική Απόκλιση 3

1.(γ')Δειγματοληψία Σπουδαιότητας (Importance Sampling) 4

1.(δ')Bootstrap Εκτιμητής 8

2. **9**

2.(α')Μέθοδος Αντιστροφής (Inversion Sampling) 9

2.(β')Μέθοδος Απόρριψης (Rejection Sampling) 10

2.(γ')Kernel Density Estimation - Cross Validated Likelihood 12

2.(δ')Bootstrap Έλεγχος - Διάστημα Εμπιστοσύνης 16

3. **18**

3.(α')Γάμμα Κατανομή - Επάρκεια - Maximum Likelihood Estimation . . 18

3.(β')Polya Κατανομή - Expectation Maximization 20

4. **26**

4.(α')Πολλαπλή Γραμμική Παλινδρόμηση - Επιλογή Μεταβλητών 26

4.(β')Μεθοδολογία Lasso 30

1.

Έστω $\phi(x) = (2\pi)^{-1/2} \exp(-x^2/2)$ η σ.π.π. της τυποποιημένης κανονικής κατανομής.

Θεωρήστε το ολοκλήρωμα:

$$J = \int_{-\infty}^{\infty} (x+a)^2 \phi(x) dx = 1 + a^2 \tag{1}$$

1.(α') Monte Carlo Ολοκλήρωση

Εκτιμήστε το J με χρήση Monte Carlo ολοκλήρωσης, προσομοιώνοντας τιμές από την τυποποιημένη κανονική κατανομή. Χρησιμοποιήστε 100 και 1000 προσομοιωμένες τιμές και θεωρήστε ότι το a = 0, 1, 2, 3, 4.

Από την Monte Carlo ολοκλήρωση γνωρίζουμε πως αν $\phi(x)$ μια συνάρτηση πυκνότητας πιθανότητας, $f(x)$ μια άλλη συνάρτηση και $x_1, x_2, ..., x_n$ ένα τυχαίο δείγμα από την $\phi(x)$ τότε:

$$\mathbb{E}[f(x)] = \int f(x)\phi(x)dx = \frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

Επομένως στο δικό μας πρόβλημα, θέτοντας $f(x) = (x+a)^2$, για ένα τυχαίο δείγμα από την $\phi(x)$ της τυποιημένης κανονικής κατανομής, έχουμε για το J:

$$\hat{J} = \frac{1}{n}\sum_{i=1}^{n} f(x_i) = \frac{1}{n}\sum_{i=1}^{n} (x_i + a)^2$$

Επομένως, ο αντίστοιχος κώδικας σε R είναι:

```
library(data.table)
MCint <- function(a, runs){
  sims <- c(rnorm(runs)) #sample from N(0,1)
  const <- 1/sqrt(2*pi)
  int <- (1/length(sims))*sum((sims+a)^2)
  return(int)
}
runs100 <- list();runs1000 <- list()
a <- c(0:4)
for(i in a){
  runs100[[i+1]] <- MCint(i,100)
  runs1000[[i+1]] <- MCint(i, 1000)
}
dt <- data.table(a,runs100,runs1000)
```

Και τα αποτελέσματα για 100 και 1000 προσωμοιώσεις, για όλες τις τιμές a:

a	$\hat{J}_{MC}^{n=100}$	$\hat{J}_{MC}^{n=1000}$	real
0	0.92	0.99	1.00
1	2.22	1.96	2.00
2	5.94	4.71	5.00
3	9.63	10.01	10.00
4	16.22	17.24	17.00

Πίνακας 1: `Monte Carlo Integration`

1.(β′) Monte Carlo Εκτιμητής - Αμεροληψία - Τυπική Απόκλιση

Αποδείξτε θεωρητικά ότι ο παραπάνω Monte Carlo εκτιμητής είναι αμερόληπτος και βρείτε την θεωρητική του τυπική απόκλιση.
• Για να δείξουμε ότι ο Monte Carlo εκτιμητής είναι αμερόληπτος αρκεί να δείξουμε ότι $\mathbb{E}[\hat{J}] = J$. Ας το υπολογίσουμε:

$$\begin{aligned}\mathbb{E}[\hat{J}] &= \frac{1}{n}\sum_{i=0}^{n}\mathbb{E}[f(x_i)] \\ &= \frac{1}{n}\cdot n\cdot\mathbb{E}[f(x)] \\ &= \int f(x)\phi(x)dx = J\end{aligned}$$

όπου στο δεύτερο βήμα της εξίσωσης εφαρμόσαμε το γεγονός το ότι η μέση τιμή είναι ένας απλός αριθμός επομένως θεωρείται σταθερό μέσα στο άθροισμα, άρα αυτό μας δίνει: $\sum_{i=0}^{n} c = n\cdot c$. Επομένως, ο Monte Carlo εκτιμητής είναι αμερόληπτος.

• Για να υπολογίσουμε την τυπική απόκλιση $\sigma(\hat{J})$, αρκεί να υπολογίσουμε το $\mathrm{Var}(\hat{J})$ και να πάρουμε τη ρίζα του.

$$\begin{aligned}\mathrm{Var}(\hat{J}) &= \mathrm{Var}\left(\frac{1}{n}\sum_{i=0}^{n}(x_i+a)^2\right) = \frac{1}{n^2}\mathrm{Var}\left(\sum_{i=0}^{n}(x_i+a)^2\right) \\ &= \frac{1}{n^2}\sum_{i=0}^{n}\mathrm{Var}\left(x_i^2+2ax_i+\cancel{a^2}\right) \\ &= \frac{1}{n^2}\sum_{i=0}^{n}\Big[\cancelto{2}{\mathrm{Var}(x_i^2)}+4a^2\cancelto{1}{\mathrm{Var}(x_i)}+4a\cdot\cancelto{0}{\mathrm{Cov}(x_i^2,x_i)}\Big] \\ &= \frac{1}{n^2}\cdot n\cdot(4a^2+2) = \frac{1}{n}(4a^2+2)\end{aligned}$$

Εδώ μένει να εξηγήσουμε τους τρεις όρους:

- $\mathrm{Var}(X^2) = 2$ όπου $X \sim \mathcal{N}(0,1)$ όπως στο πρόβλημα μας, αφού $\mathrm{Var}(X^2) = \mathbb{E}[X^4] - \left(\mathbb{E}[X^2]\right)^2 = 3 - 1 = 2$, όπου ισχύει $\mathbb{E}[X^4] = 3$, το οποίο προκύπτει εύκολα από το ολοκλήρωμα αφού γράφοντας το x^4 ως $x^3 \cdot x$ εμφανίζεται η παράγωγος του εκθετικού και μετά από μια παραγοντική ολοκλήρωση προκύπτει ένας όρος που κάνει 0 και το ολοκλήρωμα $3 \cdot J(\alpha = 0) = 3$, ενώ το $\mathbb{E}[X^2]$ εύκολα υπολογίζεται από τη σχέση $\mathrm{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$ όπου όλα είναι γνωστά.
- $\mathrm{Var}(X) = \sigma^2 = 1$.
- $\mathrm{Cov}(X, X^2) = 0$, από τον ορισμό $\mathrm{Cov}(X,Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$ μας κάνει $\mathrm{Cov}(X, X^2) = \mathbb{E}[X^3] - \mathbb{E}[X]\mathbb{E}[X^2] = 0$ καθώς $\mathbb{E}[X] = \mu = 0$ και $\mathbb{E}[X^3] = 0$ διότι η ροπή τρίτης τάξης μιας κεντραρισμένης κανονικής κατανομής είναι μηδέν, αλλά είναι εμφανές και από το ολοκλήρωμα $\mathbb{E}[X^3] = \int_{-\infty}^{\infty} x^3 \cdot \phi(x)dx$ καθώς είναι γινόμενο άρτιας με περιττής συνάρτησης, δηλαδή περιττή συνάρτηση, η οποία ολοκληρώνεται σε συμμετρικό διάστημα, επομένως είναι μηδενικό λόγω συμμετρίας.

Άρα η τυπική απόκλιση:

$$\sigma(\hat{J}) = \sqrt{\frac{4a^2 + 2}{n}}$$

1.(γ΄) Δειγματοληψία Σπουδαιότητας (Importance Sampling)

Εφαρμόστε δειγματοληψία σπουδαιότητας με χρήση της συνάρτησης $g(x) = \phi(x - a)$. Επαναλάβετε τα ερωτήματα (α) και (β) και προβείτε σε συγκρίσεις των τυπικών σφαλμάτων των δύο εκτιμητών.

Έστω η σ.π.π. $g(x) = \phi(x-a) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2}}$, και η συνάρτηση $\mathcal{Y}(x) = \frac{f(x)\cdot\phi(x)}{g(x)}$. Τότε για το ολοκλήρωμα J έχουμε:

$$J = \int f(x)\phi(x)dx = \int \frac{f(x)\cdot\phi(x)}{g(x)} \cdot g(x)dx = \int \mathcal{Y}(x)g(x)dx = \mathbb{E}_g[\mathcal{Y}(x)] = J_g$$

Δηλαδή από εκεί που το J εξέφραζε την μέση τιμή της $f(x)$ ως προς την σ.π.π. $\phi(x)$, τώρα εκφράζει την μέση τιμή της $\mathcal{Y}(x)$ ως προς την σ.π.π. $g(x)$. Ομοίως με πριν, ορίζουμε τον εκτιμητή μας:

$$\hat{J}_g = \frac{1}{n}\sum_{i=0}^{n} \mathcal{Y}(x_i) = \frac{1}{n}\sum_{i=0}^{n} (x_i + a)^2 \cdot e^{\frac{1}{2}(a^2 - 2ax_i)}$$

Ομοίως με το ερώτημα (α), ο αντίστοιχος κώδικας σε R με τα αποτελέσματά του για 100 και 1000 προσομοιωμένες τιμές, είναι:

```r
MCintIS <- function(a, runs){
  sims <- c(rnorm(runs, mean = a, sd =1))#sample from g(x) = phi(x-a)
  const <- 1/sqrt(2*pi)
  int <- (1/length(sims))*sum(exp((a^2-2*a*sims)/2)*(sims+a)^2)
  return(int)
}

ISsims_100 <- list();ISsims_1000 <- list(); real <- list()
a <- c(0:4)
for(i in a){
  ISsims_100[[i+1]] <- signif(MCintIS(i,100),4)
  ISsims_1000[[i+1]] <- signif(MCintIS(i, 1000),4)
  real[[i+1]] <- 1+i^2
}
ISdt <- data.table(a,ISsims_100,ISsims_1000,real)
```

a	$\hat{J}_{IS}^{n=100}$	$\hat{J}_{IS}^{n=1000}$	real
0	1.10	0.98	1.00
1	2.01	1.98	2.00
2	5.37	5.38	5.00
3	4.15	8.66	10.00
4	9.40	3.99	17.00

Πίνακας 2: `Importance Sampling`

• Θα δείξουμε, ομοίως με το ερώτημα (β) ότι ο εκτιμητής αυτός είναι αμερόληπτος.

$$\begin{aligned}\mathbb{E}[\hat{J}_g] &= \frac{1}{n}\sum_{i=0}^{n}\mathbb{E}_g[\mathcal{Y}(x_i)] \\ &= \frac{1}{n}\cdot n\cdot \mathbb{E}_g[\mathcal{Y}(x)] \\ &= \int \mathcal{Y}(x)g(x)dx = J_g\end{aligned}$$

• Όπως και στο ερώτημα (β), θα υπολογίσουμε την διασπορά του εκτιμητή, που είναι το τετράγωνο της τυπικής απόκλισης.

$$\mathrm{Var}(\hat{J}_g) = \mathrm{Var}\left(\frac{1}{n}\sum_{i=0}^{n}\mathcal{Y}(x_i)\right) = \frac{1}{n}\mathrm{Var}(\mathcal{Y}(x)) = \frac{1}{n}\left(\mathbb{E}_g[\mathcal{Y}^2] - [\mathbb{E}_g(\mathcal{Y}])^2\right)$$

Επομένως αρκεί να υπολογιστούν οι δύο μέσες τιμές.

• Η μέση τιμή $\mathbb{E}[\mathcal{Y}] = 1 + a^2$ προφανώς αυτό δε χρειάζεται να υπολογισθεί.

• Οπότε μένει ο υπολογισμός της $\mathbb{E}_g[\mathcal{Y}^2]$:
Σημείωση: Παραλείπονται τα άκρα ολοκλήρωσης, σε όλα τα ολοκληρώματα είναι από το $-\infty$ ως το $+\infty$.

$$\begin{aligned}\mathbb{E}_g[\mathcal{Y}^2] = \int [\mathcal{Y}(x)]^2 g(x)dx &= \int \left[(x+a)^2 \cdot e^{\frac{1}{2}(a^2-2ax)}\right]^2 \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2}} dx \\ &= \frac{1}{\sqrt{2\pi}} \int (x+a)^4 \cdot \exp\left(-\frac{1}{2}(x^2+2ax-a^2) + a^2 - a^2\right) dx \\ &= \frac{e^{a^2}}{\sqrt{2\pi}} \int (x+a)^4 \cdot e^{-\frac{1}{2}(x+a)^2} dx\end{aligned}$$

όπου με μια αλλαγή μεταβλητής $\chi \to x+a$, αυτό καταλήγει στην ροπή $4^{\eta\varsigma}$ τάξης της κανονικής κατανομής, που όπως είδαμε είναι ίση με 3:

$$\mathbb{E}_g[\mathcal{Y}^2] = e^{a^2} \frac{1}{\sqrt{2\pi}} \int \chi^4 \cdot e^{-\frac{1}{2}\chi^2} d\chi = e^{a^2} \cdot 3\sigma^4 = 3e^{a^2}$$

Επομένως έχουμε:

$$\mathrm{Var}(\hat{J}_g) = \frac{3e^{a^2} - (1+a)^2}{n}$$

και για την τυπική απόκλιση:

$$\sigma(\hat{J}_g) = \sqrt{\frac{3e^{a^2} - (1+a)^2}{n}}$$

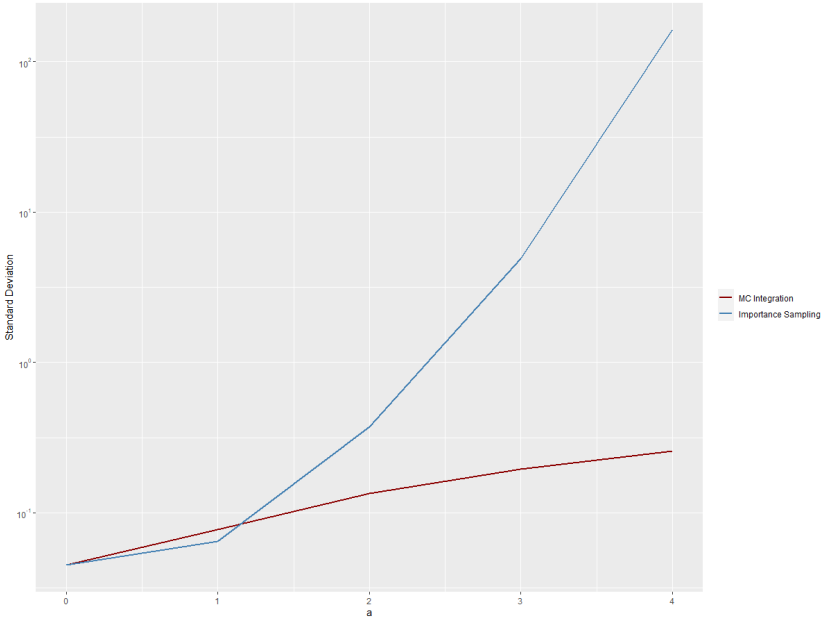
a	$\sigma(\hat{J})^{n=100}$	$\sigma(\hat{J})^{n=1000}$	$\sigma(\hat{J}_g)^{n=100}_{IS}$	$\sigma(\hat{J}_g)^{n=1000}_{IS}$
0	0.14	0.04	0.14	0.04
1	0.24	0.08	0.20	0.06
2	0.42	0.13	1.18	0.37
3	0.62	0.19	15.56	4.92
4	0.81	0.26	516.31	163.27

Πίνακας 3: `Standard Deviation for MC integration and Importance Sampling`

Ο κώδικας για το παραπάνω table φαίνεται παρακάτω.

Παρατηρούμε ότι για το Importance Sampling, για μικρές τιμές του a, δηλαδή $a = 0,1$ έχει τυπική απόκλιση συγκρίσιμη με τον εκτιμητή του (α) ερωτήματος. Καθώς όμως αυξάνει το a η τυπική απόκλιση ξεφεύγει και γίνεται αρκετά μεγαλύτερη από ότι του αρχικού εκτιμητή. Αυτό συμβαίνει διότι όπως δείξαμε, η τυπική απόκλιση εξαρτάται από έναν όρο ανάλογο του $\sqrt{3}e^{\frac{a^2}{2}}$, δηλαδή υπερεκθετικά από αυτό, το οποίο προκαλεί μια έκρηξη στο σφάλμα καθώς αυτός ο όρος μεγαλώνει πολύ απότομα. Αυτό είναι εμφανές και στο διάγραμμα 1, όπου έχουμε σχεδιάσει την τυπική απόκλιση των δύο μεθόδων για διαφορετικές τιμές του a, σε ημιλογαριθμικούς άξονες.

```r
#code to calculate standard deviation for both methods.
mc_stdev <- function(a,n){#code for MC integration SD
  return(sqrt((4*a^2+2)/n))
}
mc_rs_stdev <- function(a,n){#code for Importance Sampling SD
  numerator <- 3*exp(a^2) - (1+a^2)^2
  return(sqrt(numerator/n))
}
#instantiate lists to append
std_100 <- list(); std_1000 <- list();
std_is_100 <- list(); std_is_1000 <- list();
for(i in a){
  std_100[[i+1]] <- mc_stdev(i,100)
  std_1000[[i+1]] <- mc_stdev(i, 1000)
  std_is_100[[i+1]] <- mc_is_stdev(i,100)
  std_is_1000[[i+1]] <- mc_is_stdev(i,1000)
}
#make data table.
dt_std <- data.table(a,std_100,std_1000,std_is_100,std_is_1000)
```



Σχήμα 1: Standard Deviation vs a

Επομένως αυτή η ισχυρή εξάρτηση της τυπικής απόκλισης του εκτιμητή μας από την παράμετρο a, μας δείχνει ότι ίσως αυτή η εκλογή συνάρτησης πυκνότητας πιθανότητας για την μέθοδο Importance Sampling να μην είναι η κατάλληλη, και ότι για την προσέγγιση του ολοκληρώματος J καλύτερα να χρησιμοποιήσουμε τον απλό MC εκτιμητή.

1.(δ΄) Bootstrap Εκτιμητής

Θεωρήστε 1000 προσομοιωμένες τιμές, a = 4 και τον εκτιμητή του ερωτήματος (α). Χρησιμοποιώντας την τεχνική Bootstrap, με χρήση δικής σας συνάρτησης στην R, εκτιμήστε το τυπικό σφάλμα του εκτιμητή και συγκρίνετέ το με το θεωρητικό.

Για να υλοποιήσουμε την τεχνική Bootstrap, αρχικά προσωμοιώνουμε 1000 τιμές από την κανονική κατανομή με μέση τιμή 0 και τυπική απόκλιση 1, και θεωρούμε και τον εκτιμητή του ερωτήματος (α) για $a = 4$. Η λογική της τεχνικής αυτής είναι η δημιουργία ενός Bootstrap δείγματος, όπου από το αρχικό δείγμα τραβάμε 1000 τιμές με επανάθεση, στη συνέχεια υπολογίζουμε την τιμή του εκτιμητή με το bootstrap δείγμα, και επαναλαμβάνουμε αυτή την διαδικασία B φορές, όπου το B θεωρείται υπερπαράμετρος του προβλήματος. Ο κώδικας στην R είναι ο εξής:

```
set.seed(100) #for reproducibility.
a <- 4
B <- 50
estimator <- (rnorm(1000)+a)^2
Jhat <- c()
for(i in 1:B){
  boot_sample <- sample(estimator,size=1000,replace = TRUE)
  Jhat[i] <- sum(boot_sample)/length(boot_sample)
}
sprintf('Value of Bootstrap Estimator is: %f',mean(Jhat))
sprintf('Measure St.Dev. of Estimator is: %f',sd(Jhat))
sprintf('Theoretical St.Dev of Estimator is: %f',sqrt((4*a^2+2)/1000))
```

Όπου το output είναι:

- "Value of Bootstrap Estimator is: 17.181407".
- "Measure St.Dev. of Estimator is: 0.261588".
- "Theoretical St.Dev of Estimator is: 0.256905".

Επομένως βλέπουμε ότι έχουμε μια καλή εκτίμηση της τυπικής απόκλισης με τη μέθοδο Bootstrap, η οποία είναι της τάξης του 1.5% της μέσης τιμής. Όπως είναι αναμενόμενο, η τιμή αυτή εξαρτάται και από την υπερπαράμετρο B, την οποία θέσαμε ίση με 50.
Οι δημιουργοί της μεθόδου, B.Efron και R.J.Tibshirani, στο βιβλίο τους [1] (σελ.52) αναφέρουν ότι $B = 50$ συνήθως είναι αρκετό για τις περισσότερες εφαρμογές, ώστε να μας δώσουν μια καλή εικόνα για το σφάλμα, ενώ σπάνια απαιτείται $B > 200$. Σε μεθόδους όπως Bootstrap Confidence Intervals όπως θα δούμε στην άσκηση 2, συνήθως απαιτούνται μεγαλύτερα B, γι'αυτό και άλλωστε στην εκφώνηση δίνεται να χρησιμοποιήσουμε 1000 Bootstrap δείγματα.

2.

Έστω ότι θέλετε να προσομοιώσετε 1000 τιμές από την σ.π.π.:

$$f(x) = \frac{1}{e^3 - 1} e^x, \; x \in [0, 3] \tag{2}$$

2.(α΄) Μέθοδος Αντιστροφής (Inversion Sampling)

Χρησιμοποιήστε τη μέθοδο αντιστροφής για την προσομοίωση, με χρήση δικού σας κώδικα στην R. Συγκρίνετε το διάγραμμα της f(x) με το ιστόγραμμα των προσομοιωμένων τιμών. Καθώς η $f(x)$ είναι σ.π.π., εμείς χρειαζόμαστε την αθροιστική συνάρτηση πυκνότητας δηλαδή την C.D.F. . Αρχικά, παρατηρούμε ότι το ολοκλήρωμα της $f(x)$ στο πεδίο ορισμού της είναι μονάδα, άρα είναι «καλή» σ.π.π.. Επομένως ολοκληρώνουμε την $f(x)$ από 0 έως 3:

$$F_X(x) = \int_0^x f(x)dx = \int_0^x \frac{e^x}{e^3 - 1} dx = \frac{e^x - 1}{e^3 - 1}$$

Το ζητούμενο τώρα είναι να αντιστρέψουμε την $F_X(x)$. Παρατηρούμε ότι η $F_X(x)$ είναι γνησίως αύξουσα σε όλο το $\mathbb{R} \supseteq [0, 3]$, άρα είναι και '1-1', άρα είναι και αντιστρέψιμη. Επομένως θέτουμε $F_X(x) = y$ και έχουμε:

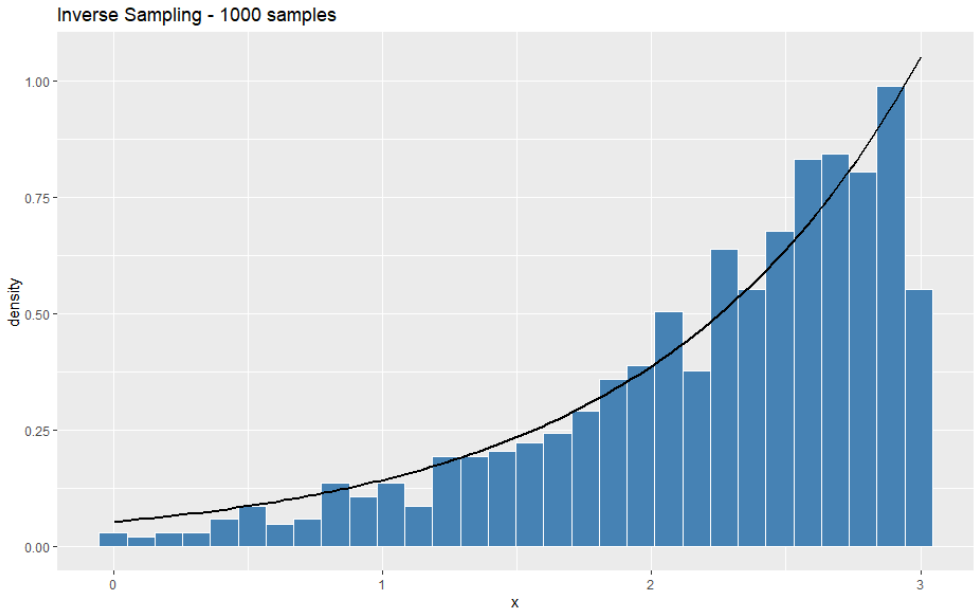
$$F_X(x) = y \Leftrightarrow e^x = y(e^3 - 1) + 1 \Leftrightarrow x = \log\left(y(e^3 - 1) + 1\right)$$

Άρα τελικά:

$$x = F_X^{-1}(u) = \log\left[u(e^3 - 1) + 1)\right]$$

Επομένως, για να προσομοιώσουμε την $f(x)$ αρκεί να παράγουμε τυχαίους αριθμούς $u \sim \mathcal{U}(0, 1)$ και να θέσουμε $x = \log\left[u(e^3 - 1) + 1)\right]$. Αυτό επιτυγχάνουμε με τον παρακάτω κώδικα, όπου δημιουργούμε το διάγραμμα 2.

```
inv_sample <- function(sims = 10^3) {
  u <- runif(sims)
  x <- log(u * (exp(3) - 1) + 1)
  return(x)
}
dt <- data.table(samples = inv_sample())
gg <- ggplot(dt, aes(x = samples)) +
  geom_histogram(aes(y = ..density..), colour = 'white', fill = 'steelblue') +
  stat_function(fun = function(x) exp(x) / (exp(3) - 1), size=0.8) + xlab("x") +
  ggtitle('Inverse Sampling - 1000 samples')
```



Σχήμα 2: Inverse Sampling - 1000 samples.

Από το σχήμα 2 βλέπουμε ότι η κατανομή που δημιουργήσαμε προσεγγίζει ικανοποιητικά την $f(x)$. Σαφώς, όσα περισσότερα δείγματα δημιουργούμε, τόσο πιο πιστά και ομαλά θα την προσεγγίζει.

2.(β΄) Μέθοδος Απόρριψης (Rejection Sampling)

Χρησιμοποιήστε την μέθοδο απόρριψης για την προσομοίωση, με χρήση δικού σας κώδικα στην R. Συγκρίνετε το διάγραμμα της f(x) με το ιστόγραμμα των προσομοιωμένων τιμών.

Θέλουμε να παράγουμε $x \sim f(x)$, το οποίο θα το πετύχουμε μέσω rejection sampling. Εισάγουμε ένα $Y \sim g(y)$ όπου $g(y)$ κατανομή εισήγησης, που στην περίπτωσή μας είναι η ομοιόμορφη κατανομή στο διάστημα $[0,1]$, πολλαπλασιασμένη με το 3 για να πάρει τη μορφή του πεδίου ορισμού της $f(x)$, άρα στο $[0,3]$. Για αυτή την κατανομή μπορούμε να βρούμε κάποιο $M > 0$ για το οποίο ισχύει: $f \leq M \cdot g = G$. Συνοπτικά ο αλγόριθμος έχει ως εξής:
• Παράγουμε $Y \sim g(y)$ και θέτουμε $y = Y$.
• Παράγουμε $u \sim \mathcal{U}(0,1)$.
• Αν $u \leq \frac{f(y)}{M \cdot g(y)}$ τότε $X = y$, αλλιώς επιστρέφουμε στην αρχή.

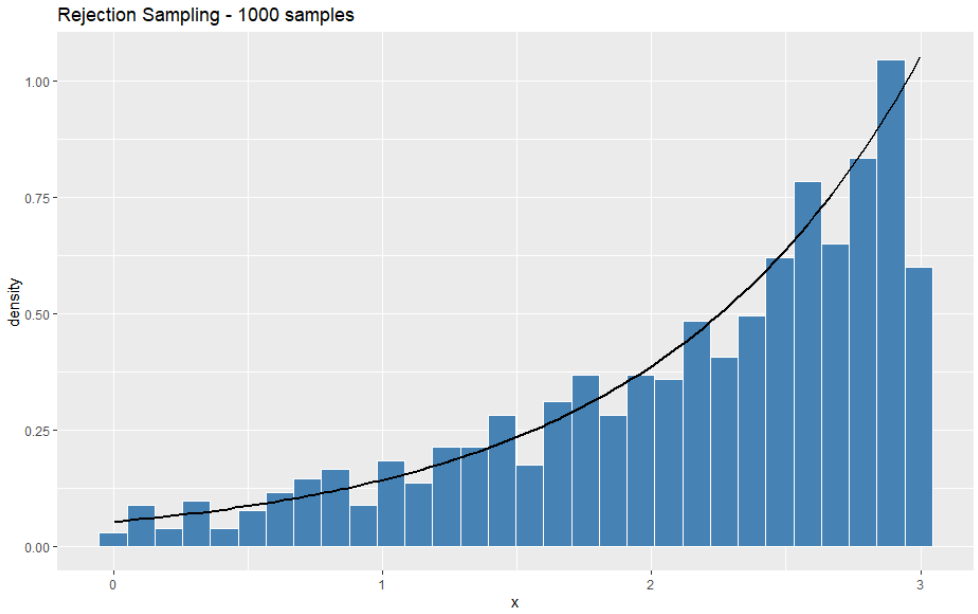
Ο αλγόριθμος αυτός δουλεύει και όταν γνωρίζουμε την μορφή της $p(x)$ σε αναλογία, απουσία της σταθεράς κανονικοποίησης, δηλαδή $p(x) = \frac{f(x)}{NC}$ όπου NC άγνωστο, γιατί μπορεί για παράδειγμα να μην μπορούμε να λύσουμε το $\int_{-\infty}^{\infty} f(x)dx$. Τότε ως M στον αλγόριθμο θα διαλέξουμε το άνω φράγμα της $f/g \leq M$. Στην δική μας περίπτωση,

```r
set.seed(97)#for reproducibility of results.
f_ofx <- function(x) { return(exp(x) / (exp(3) - 1)) }
rejection_sampling <- function(sims = 10^3) {
  accepted <- rejected <- 0
  M <- f_ofx(3)/(1/3)
  points <- c()
  while (accepted < sims) {
    #1. draw y ~ g(y) and assign y = Y
    # g(s) -> U(0,3)
    y <- runif(1, 0, 3)
    g <- dunif(y, 0, 3)
    #2. draw u ~ U(0,1) and assign u = U
    u <- runif(1, 0, 1)
    #3 if u <= f(y)/(M * g(y)) accept, else reject
    if (u <= f_ofx(y) / (M * g)) {
      accepted <- accepted + 1
      points[accepted] <- y
      print(y)
    }else { rejected <- rejected + 1 }}
  message('accepted samples:',accepted)
  message('rejected samples:',rejected)
  return(points);}
points <- data.table('samples' = rejection_sampling())
gg <- ggplot(points, aes(x = samples)) +
  geom_histogram(aes(y=..density..),colour='black',fill = 'steelblue') +
  stat_function(fun = function(x) f_ofx(x)) + xlab("x") +
  ggtitle('Rejection Sampling - 1000 samples')
  ###OUTPUT###
accepted samples:1000
rejected samples:2218
```

εφόσον η NC είναι γνωστή, διαλέγουμε $M = \sup(f(y)/g(y)) = \frac{f(3)}{1/3} = 3.157187$.

Η υλοποίηση του αλγόριθμου σε R φαίνεται στην επόμενη σελίδα, όπου έχουμε 1000 accepted samples και 2218 rejected. Αυτοί οι αριθμοί είναι απολύτως αναμενόμενοι καθώς κατά μέσο όρο θέλουμε M επαναλήψεις για να δεχτούμε ένα sample, επομένως βλέπουμε ότι για $M \sim 3$ το $1/3$ των προσομοιώσεων έχει γίνει αποδεκτό. Επίσης, στο διάγραμμα 3 συγκρίνουμε το διάγραμμα της $f(x)$ με το ιστόγραμμα των προσομοιωμένων τιμών.

Και με αυτή τη μέθοδο, στο σχήμα 3 βλέπουμε πως προσεγγίζουμε την $f(x)$ ικανοποιητικά. Αν χρησιμοποιούσαμε και εδώ παραπάνω δείγματα, η κατανομή που θα προέκυπτε θα προσέγγιζε όλο και πιο πιστά την αρχική.



Σχήμα 3: Rejection Sampling - 1000 samples.

2.(γ΄) Kernel Density Estimation - Cross Validated Likelihood

Προσομοιώστε 100 τιμές από την f(x) μέσω της μεθόδου αντιστροφής. Χρησιμοποιήστε Epanechnikov πυρήνα και τα εν λόγω δεδομένα για να εκτιμήσετε την f(x). Για την εύρεση του "βέλτιστου" πλάτους h μεγιστοποιήστε την crossvalidated πιθανοφάνεια, με χρήση δικού σας κώδικα στην R. Προβείτε σε ένα διάγραμμα της εκτιμώμενης f(x) για το h που βρήκατε και σχολιάστε το αποτέλεσμα που πήρατε.

Με την μέθοδο Kernel Density Estimation, χρησιμοποιούμε μια συνάρτηση πυρήνα (kernel function) προκειμένου να προσεγγίσουμε μια συνάρτηση πυκνότητας πιθανότητας, χρησιμοποιώντας την παρακάτω σχέση:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=0}^{n} K\left(\frac{x - x_i}{h}\right)$$

όπου $K(\cdot)$ η συνάρτηση πυρήνα, και τα δεδομένα x_i θα είναι 100 τιμές από την $f(x)$ προσομοιωμένες μέσω της μεθόδου αντιστροφής του (α) ερωτήματος. Στην δική μας περίπτωση θα χρησιμοποιήσουμε πυρήνα Epanechnikov, που έχει τη μορφή $K(x) = \frac{3}{4}(1-x^2)$, για $|x| \leq 1$. Στην πορεία, καλούμαστε να βρούμε το βέλτιστο bandwidth h. Για να το επιτύχουμε αυτό, θα χρησομοποιήσουμε την Cross Validated Maximum Likelihood. Η μορφή της εκτιμήτριας, όταν από το αρχικό δείγμα $x_1, \ldots, x_n$ αφήσουμε έξω την $i-$παρατήρηση, είναι:

$$\hat{f}_{h,-i}(x) = \frac{1}{(n-1)h} \sum_{\substack{j=1 \\ j \neq i}}^{n} K\left(\frac{x - x_j}{h}\right)$$

Η πιθανοφάνεια για το αρχικό δείγμα, δίνεται από την σχέση: $L(h) = \prod_{i=1}^{n} \hat{f}_h(x_i)$. Η Cross Validated πιθανοφάνεια όμως:

$$L(h, i) = \prod_{i=1}^{n} \hat{f}_{h,-i}(x_i)$$

Όπου αν αντικαταστήσουμε την σχέση για την εκτιμήτρια, τότε έχουμε την τελική σχέση για την Cross Validated Likelihood:

$$L_{CV}(h) = \prod_{i=1}^{n} \frac{1}{h(n-1)} \left[\sum_{\substack{j=1 \\ j \neq i}} K\left(\frac{X_i - X_j}{h} \right) \right]$$

Ο τρόπος με τον οποίο θα επιχειρήσουμε να βρούμε το h το οποίο μεγιστοποιεί την παραπάνω σχέση, είναι θεωρώντας ένα εύρος τιμών για το h (0.01 έως 5), όπου θα υπολογίζεται η πιθανοφάνεια για κάθε τιμή του, και στο τέλος θα επιστρέφεται το h^* το οποίο μεγιστοποιεί την πιθανοφάνεια. Αφού υπολογίσουμε το βέλτιστο bandwidth h^*, θα κάνουμε ένα διάγραμμα της εκτιμώμενης $f(x)$ σε σχέση με την κανονική συνάρτηση. Για να το πετύχουμε αυτό, χρειαζόμαστε μια συνάρτηση που να πραγματοποιεί το Kernel Density Estimation, δηλαδή να μας επιστρέφει την $\hat{f}(x)$. Η βοηθητική συνάρτηση αυτή φαίνεται παρακάτω:

```
set.seed(37)# for reproducibility
epanechnikov <- function(x) { return(3 / 4 * (1 - x^2)) }
# Function to return Kernel Density Estimates given a seq. x, data:xi, and bw:h.
KDE <- function(x,xi,h){
  n <- length(xi)
  const <- 1/(n*h)
  fhats <- c()
  for(i in 1:n){
    inner_sum <- 0
    for(j in 1:n){
      u <- (x[i]-xi[j])/h
      if(abs(u)<=1)
      {inner_sum <- inner_sum + epanechnikov(u)}    }
    fhats[i] <- const*inner_sum  }
return(fhats)}
```

Στην συνέχεια παραθέτουμε τον κώδικα που χρησιμοποιήθηκε προκειμένου να προσδιορίσουμε την βέλτιστη τιμή του h.

```r
L_CV <- function(xi,h_min,h_max,step){
  likelihood <- -Inf
  h_star <- 0
  h <- h_min
  n <- length(xi)
  fhat <- c()
  while(h<=h_max){
    current_like <- 0
    for(i in 1:length(xi)){
      inner <- 0
      for(j in 1:length(xi)){
        if(i!=j){
          u <- (xi[i]-xi[j])/h
          if(abs(u)<=1){
            inner <- inner + epanechnikov(u) }}}
      fhat[i] <- 1/((n-1)*h)*inner  }
  current_like <- prod(fhat)
  if (current_like > likelihood) {
    likelihood <- current_like
    h_star <- h}
  h <- h + step}
  cat("Optimal value of h is:", h_star)
  return(h_star)
}
```

Επομένως βλέπουμε ότι το βέλτιστο h που έβγαλε ο αλγόριθμος είναι η τιμή **h = 0.17**. Με τον παρακάτω κώδικα δημιουργήθηκε το δείγμα, κλήθηκαν οι συναρτήσεις, και έγινε το διάγραμμα 4.

```r
xi <- inv_sample(100)#data
x <- seq(0,3,length.out = length(xi))#seq
h_star = L_CV(xi,h_min=0.01,h_max=5,step=0.01)
kde <- KDE(x,xi,h_star)
eq <- function(x){return(ifelse(x >= 0 & x <= 3,exp(x)/(exp(3)-1),0))}
data <- data.table(x,kde,xi)
gg <- ggplot(data = data) +
  geom_line(aes(x = x, y = as.numeric(kde), color = 'darkred'),size=0.8) +
  stat_function(fun=eq,aes(color='steelblue'),size=0.75) +
  xlab(paste("N = 100, bw = ",h_star)) + ylab('density') +
  ggtitle('')+ scale_color_identity(name="",breaks=c("darkred","steelblue"),
                        labels=c("KDE - Epanechnikov","f(x)"),guide='legend')+
  geom_rug(aes(x=xi),length=unit(0.02,"npc"))+xlim(0,3.01)
```

Σχήμα 4: Kernel Density Estimation - Epanechnikov Kernel.

Η συνάρτηση πήρε περίπου 3 sec να εκτελεστεί, όμως είναι υψηλής πολυπλοκότητας επομένως για μεγαλύτερο δείγμα θα έπρεπε να εξετάσουμε τον τρόπο υλοποίησης ή να την παραλληλοποιήσουμε. Παρατηρούμε ότι η εκτίμηση που έχουμε, παρουσιάζει πολλές ταλαντώσεις γύρω από την πραγματική $f(x)$, και αυτό οφείλεται στο μικρό αριθμό δειγμάτων. Παρ'όλα αυτά είναι αντιπροσωπευτική των δειγμάτων καθώς εκεί που έχει περισσότερα δείγματα μιμείται πιο καλά την $f(x)$. Επομένως, όσο αυξάνουμε τον αριθμό των samples, τόσο πιο πιστή θα είναι η εκτίμηση.

2.(δ΄) Bootstrap Έλεγχος - Διάστημα Εμπιστοσύνης

Προσομοιώστε 10 τιμές από την f(x) μέσω της μεθόδου αντιστροφής. Χρησιμοποιώντας τις εν λόγω τιμές προβείτε σε έναν Bootstrap έλεγχο υπόθεσης (χρησιμοποιώντας δική σας συνάρτηση και όχι κάποια έτοιμη της R) της μηδενικής υπόθεσης $\mu = 2$ έναντι της εναλλακτικής $\mu \neq 2$, σε επίπεδο σημαντικότητας 5%, όπου το μ δηλώνει την (υποθετικά) άγνωστη μέση τιμή της κατανομής f(x). Απαντήστε στο ερευνητικό σας ερώτημα και με τη βοήθεια ενός 95% Bootstrap διαστήματος εμπιστοσύνης (χρησιμοποιώντας δική σας συνάρτηση και όχι κάποια έτοιμη της R), βασισμένου σε ποσοστιαία σημεία. Για τον έλεγχο υπόθεσης και για το διάστημα εμπιστοσύνης χρησιμοποιήστε 1000 Bootstrap δείγματα. Βρείτε την πραγματική μέση τιμή της $f(x)$ και σχολιάστε τα αποτελέσματα του ελέγχου και του διαστήματος εμπιστοσύνης.

Όσον αφορά τον Bootstrap έλεγχο, στην αρχή προσομοιώνουμε 10 τιμές από την $f(x)$ μέσω της μεθόδου αντιστροφής, όπως στο (α) ερώτημα, και έχουμε ένα διάνυσμα $\mathbf{x} = (x_1, \ldots, x_{10})$. Εμείς καλούμαστε να εξετάσουμε την μηδενική υπόθεση H_0, έναντι της H_1, όπου:

$$\begin{cases} H_0 : \ \mu = 2 \\ H_1 : \ \mu \neq 2 \end{cases}$$

Διαλέγουμε την ελεγχοσυνάρτηση $T = |\bar{x} - 2|$, η οποία θα είναι 0 αν ισχύει η H_0, ενώ μεγαλύτερες τιμές υποδεικνύουν ότι αποκλίνουμε από αυτή. Η ιδέα είναι να δημιουργήσουμε δείγματα από την H_0, όμως αυτό δεν ξέρουμε αν είναι εφικτό διότι η μέση τιμή $\bar{x}$ μπορεί να είναι διαφορετική, γι'αυτό «κεντράρουμε» τα δεδομένα μας στο $\mu = 2$, αφαιρώντας το T, η αλλιώς: $\tilde{x} = x - \bar{x} + 2$, όπου $\tilde{x}$ είναι τα κεντραρισμένα μας δεδομένα. Τώρα ικανοποιείται η H_0. Επομένως προσομοιώνουμε $B = 1000$ bootstrap δείγματα, και για κάθε Bootstrap δείγμα υπολογίζουμε την ελεγχοσυνάρτηση T, και έχουμε ένα δείγμα από 1000 bootstrap εκτιμήσεις, έστω $\hat{\theta} = (T_1^*, T_2^*, \ldots, T_{1000}^*)$. Ο έλεγχος της υπόθεσης θα γίνει με το p-value, που ορίζεται ως: $p = \frac{m+1}{B+1}$, όπου m είναι ο αριθμός παρατηρήσεων για τις οποίες ισχύει $\hat{\theta}_i^* > T$. Επομένως, αφού έχουμε επίπεδο σημαντικότητας $\alpha = 0.05$, αν $p > 0.05$ τότε αποδεχόμαστε την H_0, ενώ αν $p < 0.05$ την απορρίπτουμε. Η υλοποίηση του αλγορίθμου σε R βρίσκεται στην παρακάτω σελίδα.
Το output είναι: **"Null Hypothesis accepted with p_val: 0.398601"**
Είναι σημαντικό να σημειώσουμε πως με το δικό μας random seed το οποίο έχουμε εμείς κρατήσει σταθερό, η τιμή αυτή είναι πάντα ίδια, αλλά αυτό δεν ισχύει σε όλες τις περιπτώσεις καθώς η τυχαιότητα του αρχικού δείγματος το οποίο παίρνουμε με inverse sampling, περιλαμβάνει την λήψη uniform τυχαίων αριθμών, οπότε αυτή η τυχαιότητα ενδέχεται να αλλάξει το αποτέλεσμα, ειδικά όταν αυτό είναι τόσο μικρού μεγέθους.

Στην συνέχεια ζητείται η κατασκευή ενός Bootstrap διαστήματος εμπιστοσύνης, βασισμένου σε ποσοστιαία σημεία (percentile bootstrap). Δηλαδή:

$$\left(\theta^*_{\alpha/2}, \theta^*_{1-\alpha/2}\right)$$

Για να κατασκευαστεί αυτό, αρκεί να πάρουμε τις 1000 Bootstrap εκτιμήσεις που υπολογίσαμε, να τις κάνουμε sort, και να πάρουμε τις παρατηρήσεις που αντιστοιχούν σε αυτά τα quantiles. Επομένως από τις δύο τελευταίες εντολές του παραπάνω κώδικα έχουμε:
"Bootstrap confidence interval: (0.007152, 0.533819)"
"T=0.220243 falls within the Confidence Interval" Επομένως, βλέπουμε ότι το T πέφτει εντός του διαστήματος εμπιστοσύνης, επομένως δε μπορούμε παρά να δεχτούμε την $H_0: \mu = 2$. Αυτό βέβαια εξαρτάται ξανά από το random seed, αν ήταν διαφορετικό, τότε μπορεί να είχαμε διαφορετικό αποτέλεσμα. Η πραγματική μέση τιμή της συνάρτησης υπολογίζεται με απλή παραγοντική ολοκλήρωση:

$$\mu_{real} = \int_0^3 xf(x)dx = \frac{1}{e^3-1}\int_0^3 xe^x dx = \frac{1}{e^3-1}\cdot\left(xe^x\Big|_0^3 - \int_0^3 e^x dx\right) = \frac{2e^3+1}{e^3-1} \approx 2.157$$

Επομένως, παρατηρούμε ότι λανθασμένα αποδεχτήκαμε την H_0, κι αυτό οφείλεται στον πολύ μικρό αριθμό δειγμάτων, ο οποίος ήταν 10. Αν πάρουμε π.χ. 1000 δείγματα, τότε αυξάνεται η ακρίβεια και με μεγάλη βεβαιότητα, το T πέφτει έξω από το διάστημα εμπιστοσύνης.

```
#BOOTSTRAP HYPOTHESIS TESTING
set.seed(97)#for reproducibility
a < - 0.05
x <- inv_sample(10) #sample 10
x_center <- x - mean(x) + 2 #center
bstrap_test <- function(x, B = 1000){
  bootstrap <- c()
  for(i in 1:B){
    x_boot <- sample(x,size=length(x), replace = TRUE)
    bootstrap <- c(bootstrap,abs(mean(x_boot)-2))
  }
  return(bootstrap)
}
thetahat <- bstrap_test(x_center, B = 1000)
pval <- (sum(thetahat > T) + 1)/1001
if(pval > a) {
  sprintf('Null Hypothesis accepted with p_val: %f',pval)
}else{
  sprintf('Null Hypothesis rejected with p_val: %f',pval)
}
#BOOTSTRAP CONFIDENCE INTERVAL
sorted <- sort(thetahat)
T <- abs(mean(x) - 2)#calculate T from initial sample
sprintf('Bootstrap confidence interval: (%f, %f)',sorted[25],sorted[975])
if(T > sorted[25] & T < sorted[975]){
      sprintf('T=%f falls within the Confidence Interval',T)
}else{sprintf('T=%f falls outside the Confidence Interval',T)}
```

3.

3.(α΄) Γάμμα Κατανομή - Επάρκεια - Maximum Likelihood Estimation

Θεωρήστε την Γάμμα κατανομή με σ.π.π.: $f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$, όπου $\alpha, \beta > 0$ άγνωστες παράμετροι. Έστω ότι διαθέτετε τυχαίο δείγμα μεγέθους n από την παραπάνω κατανομή. Ποια είναι η επαρκής στατιστική συνάρτηση και τι διάσταση έχει; Αναπτύξτε θεωρητικά (μόνο) τα βήματα του Newton Raphson αλγόριθμου για την μεγιστοποίηση της λογαριθμικής πιθανοφάνειας ως προς α και β.
• Για να βρούμε τις επαρκείς στατιστικές, θα χρειαστούμε ένα θεώρημα, το οποίο παραθέτουμε χωρίς να το αποδείξουμε. Η απόδειξη βρίσκεται στο βιβλίο [2] (θεώρημα 6.1.1.).

Παραγοντικό κριτήριο των Neyman–Fisher: Έστω ότι το δείγμα X έχει πυκνότητα $f(x;\theta)$, $\theta \in \Theta$. Τότε η στατιστική συνάρτηση $T(X)$ είναι επαρκής εάν και μόνο αν υπάρχουν μη αρνητικές συναρτήσεις q και h, με την h να μην εξαρτάται από το θ, έτσι ώστε:

$$f(x;\theta) = q\left(T(x),\theta\right) \cdot h(x), \ \forall x, \ \forall \theta$$

Δηλαδή, η σ.π.π. μπορεί να παραγοντοποιηθεί σε ένα γινόμενο όπου ο ένας παράγοντας (ο h) δεν εξαρτάται από το θ, και ο άλλος παράγοντας (ο q) εξαρτάται από το x μόνο μέσω της στατιστικής συνάρτησης $T(x)$. Αν το δείγμα μας είναι τυχαίο σημαίνει ότι οι παρατηρήσεις είναι ανεξάρτητες και κατανεμημένες κατά την κατανομή Γάμμα (α,β). Επομένως, για το δείγμα $X = (x_1, x_2, \ldots, x_n)$ μπορούμε να πάρουμε την από κοινού συνάρτηση πυκνότητας πιθανότητας, και να την γράψουμε ως:

$$\begin{aligned} f(x_1,\ldots,x_n;\alpha,\beta) &= \prod_{i=1}^{n} \left(\frac{\beta^\alpha}{\Gamma(\alpha)}\right) x_i^{\alpha-1} e^{-\beta x_i} \\ &= \left(\frac{\beta^\alpha}{\Gamma(\alpha)}\right)^n \cdot \prod_{i=1}^{n} x_i^{\alpha-1} e^{-\beta x_i} \\ &= \left(\frac{\beta^\alpha}{\Gamma(\alpha)}\right)^n \cdot \left(\prod_{i=1}^{n} x_i\right)^{\alpha-1} \cdot \left(e^{-\beta \sum_{i=1}^{n} x_i}\right) \\ &= h(x_i) \cdot q(T(x_i),\alpha,\beta) \end{aligned}$$

όπου έχουμε θέσει $h(x) = 1$, και η $q(x,\alpha,\beta)$ εξαρτάται μόνο από το $T(x)$, αν θέσουμε $T(x_i) = \left(\prod_{i=1}^{n} x_i, \sum_{i=1}^{n} x_i\right)$.

Επομένως, ακολουθώντας το θεώρημα Neyman-Fisher, η συνάρτηση:

$$T(x_i) = \left(\prod_{i=1}^{n} x_i, \sum_{i=1}^{n} x_i\right)$$

είναι επαρκής στατιστική συνάρτηση των (α, β), με διάσταση 2.
Αν κάναμε το αντίστοιχο με την λογαριθμημένη πιθανοφάνεια, όπως είδαμε στο μάθημα, αντί για τον όρο $\prod_{i=1}^{n} x_i$, θα είχαμε τον όρο $\sum_{i=1}^{n} \log(x_i)$.
• Η συνάρτηση πιθανοφάνειας, έχει τη μορφή που υπολογίσαμε πριν:

$$\mathcal{L}(\alpha, \beta) = \left(\frac{\beta^\alpha}{\Gamma(\alpha)}\right)^n \cdot \left(\prod_{i=1}^{n} x_i\right)^{\alpha-1} \cdot \left(e^{-\beta \sum_{i=1}^{n} x_i}\right)$$

Όπου αν την λογαριθμήσουμε, παίρνουμε την λογαριθμική πιθανοφάνεια:

$$\ell(\alpha, \beta) = n\alpha \log \beta - n \log \Gamma(\alpha) + (\alpha - 1) \sum_{i=1}^{n} \log x_i - \beta \sum_{i=1}^{n} x_i$$

Για να τη μεγιστοποιήσουμε ως προς α και β, παίρνουμε τις μερικές παραγώγους ως προς αυτά, λαμβάνοντας υπόψη πως η παράγωγος ως προς β έχει εξάρτηση ως προς α.

$$\frac{\partial \ell(\alpha, \beta)}{\partial \beta} = \frac{n\alpha}{\beta} - \sum_{i=1}^{n} x_i = 0 \Leftrightarrow \beta = \frac{\alpha}{\bar{x}}$$

$$\frac{\partial \ell(\alpha, \beta)}{\partial \alpha} = n\log(\frac{\alpha}{\bar{x}}) - n(\log\Gamma(\alpha))' + \sum_{i=1}^{n} \log(x_i) = 0$$

Επομένως αν θέλουμε να βρούμε τα α και β που μεγιστοποιούν την λογαριθμική πιθανοφάνεια, αρκεί να λύσουμε τις παραπάνω εξισώσεις. Βλέπουμε ότι $\beta = \alpha/\bar{x}$, επομένως αν καταφέρουμε να βρούμε μια λύση για την εξίσωση του α, είναι εύκολο να υπολογίσουμε το β. Εκ' πρώτης όψεως δεν είναι τεριμμένη η εύρεσης μιας λύσης κλειστής μορφής, γι'αυτό μπορούμε να χρησιμοποιήσουμε την μέθοδο Newton-Raphson που είναι μέθοδος αριθμητικής επίλυσης εξισώσεων. Αν θέλουμε για παράδειγμα να βρούμε την ρίζα μιας συνάρτησης $f(x)$, τότε χρησιμοποιείται ο αναδρομικός τύπος:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Εμείς θέλουμε να υπολογίσουμε την ρίζα της συνάρτησης $\ell'_\alpha \equiv \frac{\partial \ell(\alpha,\beta)}{\partial \alpha}$, όπου υιοθετούμε αυτό το συμβολισμό για ευκολία. Πριν γράψουμε τον αναδρομικό τύπο για την επίλυση της εξίσωσης αυτής, θα ορίσουμε κάποιες χρήσιμες συναρτήσεις που θα μας βοηθήσουν.
Είδαμε παραπάνω ότι στην εξίσωση εμφανίζεται η παράγωγος του $\log(\Gamma(\alpha))$. Αυτός είναι ο ορισμός της δι-γάμμα συνάρτησης, δηλαδή: $\psi(\alpha) = \frac{d}{d\alpha}\log(\Gamma(\alpha))$. Επειδή θα χρειαστούμε και την παράγωγο αυτής της συνάρτησης, αυτή ονομάζεται τρι-γάμμα συνάρτηση και ορίζεται αντίστοιχα ως: $\psi_1(\alpha) = \frac{d^2}{d\alpha^2}\log(\Gamma(\alpha)) = \frac{d}{d\alpha}\psi(\alpha)$. Ο αναδρομικός τύπος λοιπόν της μεθόδου Newton-Raphson είναι $\alpha_{n+1} = \alpha_n - \frac{\ell'_\alpha(\alpha_n)}{\ell''_\alpha(\alpha_n)}$, όπου αντικαθιστώντας και απλοποιώντας τις εκφράσεις έχουμε:

$$\alpha_{n+1} \leftarrow \alpha_n - \frac{\psi(\alpha_n) - \log(\frac{\alpha}{\bar{x}}) - \frac{1}{n}\sum_{i=0}^{n} \log(x_i)}{\psi_1(\alpha_n) - \frac{1}{a_n}}$$

3.(β΄) Polya Κατανομή - Expectation Maximization

Η κατανομή Polya(α, β) αποτελεί μια γενίκευση της Αρνητικής Διωνυμικής. Εξαρτάται από δύο παραμέτρους $\alpha, \beta > 0$ και έχει σ.π.π.:

$$f(x) = \frac{\Gamma(x+\alpha)}{x!\Gamma(\alpha)} \left(\frac{\beta}{1+\beta}\right)^{\alpha} \left(\frac{1}{1+\beta}\right)^{x}.$$

Αρχικά αποδείξτε ότι η τ.μ. X ακολουθεί την κατανομή Polya(α, β) όταν:

$$X|\theta \sim \text{Poisson}(\theta) \text{ και } \theta \sim \text{Gamma}(\alpha, \beta).$$

Επιλέξτε $\alpha = 2$ και $\beta = 5$ και με χρήση της παραπάνω μίξης προσομοιώστε $n = 10000$ τιμές x_i από την Polya(α, β) κατανομή. Χωρίς να αποθηκεύσετε τις θ_i τιμές θεωρήστε πως τα «πλήρη» δεδομένα σας είναι τα ζεύγη (x_i, θ_i), $i = 1, \ldots, 10000$, όπου θ_i είναι «ελλιπείς» τιμες. Σκοπος σας είναι να εκτιμήσετε τις παραμέτρους α και β. Αν είχατε παρατηρήσει τα θ_i τότε θα έπρεπε απλά να εκτιμήσετε τις παραμέτρους της κατανομής Γάμμα, με την μέθοδο μεγίστης πιθανοφάνειας. Θεωρήστε τον αλγόριθμο EM για την εκτίμηση του α και β. Αναπτύξτε (θεωρητικά) πλήρως τα βήματα του αλγορίθμου, με χρήση των επαρκών στατιστικών, και εν συνεχεία δημιουργήστε μια δική σας συνάρτηση στην R που θα υλοποιεί τον αλγόριθμο. Στο M-step θα χρειαστεί να εφαρμόστε τον αλγόριθμο Newton-Raphson του ερωτήματος (α). Επίσης θα χρειαστείτε τις συναρτήσεις digamma και trigamma της R. Ως αρχικές τιμές θεωρήστε τις $\alpha = 1$ και $\beta = 1$ και ως κριτήριο τερματισμού, για δύο διαδοχικές επαναλήψεις (r) και (r+1), χρησιμοποιήστε:

$$\left(\alpha^{(r+1)} - \alpha^{(r)}\right)^2 + \left(\beta^{(r+1)} - \beta^{(r)}\right)^2 \leq 10^{-10}.$$

Πόσο καλά ο αλγόριθμος εκτίμησε τις τιμές των α και β;

• Αρχικά για το αποδεικτικό σκέλος, πρέπει να αποδείξουμε ότι η κατανομή Polya μπορεί να οριστεί σαν μια κατανομή χτισμένη από ιεραρχία, δηλαδή αν μια τυχαία μεταβλητή X ακολουθεί την κατανομή Poisson με παράμετρο θ, και σ.π.π.:

$$f_{\text{poisson}}(x \mid \theta) = \frac{\theta^x \cdot e^{-\theta}}{x!}$$

και η παράμετρος Poisson ακολουθεί με τη σειρά της την κατανομή Γάμμα(α,β), με σ.π.π.:

$$f_{\text{gamma}}(\theta; \alpha, \beta) = \frac{\beta^{\alpha} \cdot \theta^{\alpha-1}}{\Gamma(\alpha)} e^{-\beta\theta}$$

τότε η X ακολουθεί την κατανομή Polya με σταθερές α, β.
Αυτό και θα αποδείξουμε, έχουμε:

$$P(X = x) = P(X = x, 0 < \theta < \infty) = \int_0^\infty f(x, \theta) d\theta$$

$$= \int_0^\infty f(x \mid \theta) f(\theta) d\theta = \int_0^\infty \left(\frac{\theta^x \cdot e^{-\theta}}{x!} \right) \left(\frac{\beta^\alpha \cdot \theta^{\alpha-1}}{\Gamma(\alpha)} e^{-\beta\theta} \right) d\theta$$

$$= \frac{\beta^\alpha}{\Gamma(\alpha)x!} \int_0^\infty \theta^{x+\alpha-1} e^{-\theta(\beta+1)} d\theta \overset{y=\theta(\beta+1)}{=} \frac{\beta^\alpha}{\Gamma(\alpha)x!} \int_0^\infty \left(\frac{y}{\beta+1} \right)^{x+\alpha-1} \cdot e^{-y} \frac{dy}{\beta+1}$$

$$= \frac{\beta^\alpha}{\Gamma(\alpha)x!} \left(\frac{1}{\beta+1} \right)^x \left(\frac{1}{\beta+1} \right)^\alpha \int_0^\infty y^{x+\alpha-1} e^{-y} dy$$

$$= \frac{1}{\Gamma(\alpha)x!} \left(\frac{1}{\beta+1} \right)^x \left(\frac{\beta}{\beta+1} \right)^\alpha \cdot \Gamma(x+\alpha)$$

$$= \frac{\Gamma(x+\alpha)}{\Gamma(\alpha)x!} \left(\frac{1}{\beta+1} \right)^x \left(\frac{\beta}{\beta+1} \right)^\alpha = \mathbf{f_{Polya}}(\mathbf{x}, \boldsymbol{\theta}; \boldsymbol{\alpha}, \boldsymbol{\beta})$$

Επομένως η τ.μ. X ακολουθεί κατανομή Polya με παραμέτρους α, β.

• Για το κομμάτι του EM, σκοπός μας είναι να εκτιμήσουμε τις παραμέτρους α, β χωρίς να έχουμε παρατηρήσει τα latent data θ_i της poisson. Επομένως σκοπός μας είναι να μεγιστοποιήσουμε την πιθανοφάνεια: $\max_\theta(\mathcal{L}(\theta \mid x))$. Επομένως για το Likelihood, με χρήση του N.Bayes: $P(x_k, \theta_k; \alpha, \beta) = P(x_k \mid \theta_x; \alpha, \beta) \cdot P(\theta_k; \alpha, \beta)$, έχουμε:

$$\mathcal{L}(\alpha, \beta) = \sum_{k=1}^{n} \log \left[P(x_k \mid \theta_x; \alpha, \beta) \cdot P(\theta_k; \alpha, \beta) \right]$$

Επομένως, για το **E-step**, ξεκινόντας από κάποιες αρχικές τιμές $\theta^{(0)} = (\alpha_0, \beta_0)$, μέχρι κάποια επανάληψη $\theta^{(r)} = (\alpha, \beta)$. Εδώ το θ δεν είναι οι latent variables μας, απλά χρησιμοποιείται προσωρινά ο συμβολισμός αυτός των παραμέτρων για να συνάδει με τις σημειώσεις.

$$Q(\theta^{(0)}, \theta^{(r)}) = Q(\alpha_0, \beta_0; \alpha, \beta) = \mathbb{E}_{\theta_k} \left[\sum_{k=1}^{n} \log \left(P(x_k \mid \theta_x; \alpha, \beta) \cdot P(\theta_k; \alpha, \beta) \right) \right]$$

$$= \sum_{k=1}^{n} \mathbb{E}_{\theta_k} \Big[\log \left[P(x_k \mid \theta_k; a, \beta) \cdot P(\theta_k; \alpha, \beta) \right] \Big]$$

$$= \sum_{k=1}^{n} \int_0^\infty \underbrace{P(\theta_k \mid x_k; \alpha_0, \beta_0)}_{①} \cdot \underbrace{\log \left[P(x_k \mid \theta_k; a, \beta) \cdot P(\theta_k; \alpha, \beta) \right]}_{②} d\theta_k$$

Όπου θα υπολογίσουμε ξεχωριστά τις ποσότητες ① και ②.

Για το $\textcircled{1}$, μέσω του θ. Bayes έχουμε:

$$\begin{aligned}\textcircled{1} = P(\theta_k \mid x_k; \alpha_0, \beta_0) &= \frac{P(x_k \mid \theta_k; a_0, \beta_0) \cdot P(\theta_k; a_0, \beta_0)}{P(x_k)} = \frac{\frac{\theta_k^{x_k} e^{-\theta_k}}{\cancel{x_k!}} \frac{\cancel{\beta_0^{\alpha_0}}}{\cancel{\Gamma(\alpha_0)}} \theta_k^{a_0-1} e^{-\beta_0\theta_k}}{\frac{\Gamma(x_k+\alpha_0)}{\cancel{x_k!}\cancel{\Gamma(\alpha_0)}} \left(\frac{1}{1+\beta_0}\right)^{x_k} \left(\frac{\cancel{\beta_0}}{1+\beta_0}\right)^{\alpha_0}} \\ &= \frac{e^{-\theta_k(1+\beta_0)} \cdot \theta_k^{x_k+\alpha_0-1} \cdot (1+\beta_0)^{x_k+\alpha_0}}{\Gamma(x_k+\alpha_0)} = \mathrm{Gamma}_{(x_k+\alpha_0, 1+\beta_0)}(\theta_k)\end{aligned}$$

δηλαδή αυτός ο όρος ακολουθεί την γάμμα κατανομή με σταθερές τις $x_k + \alpha_0$ και $1 + \beta_0$, που σημαίνει ότι ολοκληρώνεται στην μονάδα, καθώς είναι σ.π.π., αυτό θα μας χρειαστεί παρακάτω.

Για τον όρο $\textcircled{2}$ έχουμε:

$$\begin{aligned}\textcircled{2} &= \log\left[P\left(x_k \mid \theta_k; a, \beta\right) \cdot P(\theta_k; \alpha, \beta)\right] = \log\left[\frac{\theta^x \cdot e^{-\theta}}{x!} \cdot \frac{\beta^\alpha \cdot \theta^{\alpha-1}}{\Gamma(\alpha)} \cdot e^{-\beta\theta}\right] \\ &= \log\left[\frac{\theta_k^{x_k+\alpha-1} \cdot \beta^\alpha \cdot e^{-\theta_k(\beta+1)}}{x_k! \cdot \Gamma(\alpha)}\right] \\ &= (x_k + \alpha - 1)\log(\theta_k) + \alpha\log(\beta) - \theta_k(1+\beta) - \sum_{k=1}^{n} \log(x_k) - \log(\Gamma(a))\end{aligned}$$

Επομένως, επιστρέφοντας στον υπολογισμό του $\mathcal{Q}$:

$$\begin{aligned}Q(\alpha_0, \beta_0; \alpha, \beta) &= \sum_{k=1}^{n} \int_0^\infty \mathrm{Gamma}_{(x_k+\alpha_0, 1+\beta_0)}(\theta_k) \Big[(x_k + \alpha - 1)\log(\theta_k) + \alpha\log(\beta) - \theta_k(1+\beta) \\ &\quad - \sum_{k=1}^{n} \log(x_k) - \log(\Gamma(a))\Big] d\theta_k \\ &= -n\sum_{k=1}^{n} \log(x_k) - n\log(\Gamma(a)) + n\alpha\log(\beta) - \sum_{k=1}^{n} (1+\beta)\mathbb{E}\left[\mathrm{Gamma}_{(x_k+\alpha_0, 1+\beta_0)}\right] \\ &\quad + \sum_{k=1}^{n} \int_0^\infty \mathrm{Gamma}_{(x_k+\alpha_0, 1+\beta_0)} \cdot (x_k + \alpha - 1)\log(\theta_k) d\theta_k \\ &= -n\sum_{k=1}^{n} \log(x_k) - n\log(\Gamma(a)) + n\alpha\log(\beta) - \sum_{k=1}^{n} (1+\beta)\frac{x_k+\alpha_0}{1+\beta_0} \\ &\quad + \sum_{k=1}^{n} (x_k + \alpha - 1)\mathbb{E}_{\theta_k|x_k}\Big[\log(\theta_k)\Big]\end{aligned}$$

Όπου χρησιμοποιήθηκε το γεγονός ότι αν μια τ.μ. ακολουθεί την $\mathrm{Gamma}(\alpha, \beta)$ τότε $\mathbb{E}[\mathrm{Gamma}(\alpha, \beta)] = \frac{\alpha}{\beta}$, και στην πορεία θα χρειαστούμε και την ιδιότητα

$\mathbb{E}[\log(Gamma(\alpha,\beta))] = \psi(\alpha) - \log(\beta)$, οι οποίες έχουν αποδειχθεί στο μάθημα. Η τελική τιμή για το Q είναι:

$$\begin{aligned} Q(\alpha_0,\beta_0;\alpha,\beta) = &-n\sum_{k=1}^{n}\log(x_k) - n\log(\Gamma(a)) + n\alpha\log(\beta) - \sum_{k=1}^{n}(1+\beta)\frac{x_k+\alpha_0}{1+\beta_0} \\ &+ \sum_{k=1}^{n}(x_k+\alpha-1)\left[\psi(x_k+\alpha_0) - \log(1+\beta_0)\right] \end{aligned} \tag{3}$$

Επομένως, για να μεγιστοποιήσουμε το Q, πρέπει να υπολογίσουμε τις μερικές του παραγώγους, αρχικά ως προς β:

$$\frac{\partial Q}{\partial \beta} = \frac{n\alpha}{\beta} - \frac{n(\bar{x}+\alpha_0)}{1+\beta_0} = 0 \Rightarrow \beta = \frac{\alpha(1+\beta_0)}{\bar{x}+\alpha_0}$$

Αφού το β εξαρτάται από το α πρέπει να λάβουμε υπόψιν και την παράγωγο $\frac{\partial\beta}{\partial\alpha} = \frac{1+\beta_0}{\bar{x}+\alpha_0}$, έπειτα ως προς α:

$$\frac{\partial Q}{\partial \alpha} = -n\psi(\alpha) + n\log\left[\frac{\alpha}{\bar{x}+\alpha_0}\right] + \sum_{k=1}^{n}[\psi(x_k+\alpha_0)] = 0$$

Όπου εκ πρώτης όψεως δεν έχει κλειστής μορφής λύση, επομένως θα κάνουμε ότι και στο ερώτημα (α), θα εφαρμόσουμε την μέθοδο Newton Raphson για να βρούμε τη λύση της. Επομένως υπολογίζουμε και την δεύτερη παράγωγο ως προς α:

$$\frac{\partial^2 Q}{\partial \alpha^2} = -n\psi_1(\alpha) + n\frac{\bar{x}+\alpha_0}{\alpha(1+\beta_0)}$$

Άρα, για το **M-step**, έχουμε τον εξής κανόνα ανανέωσης:

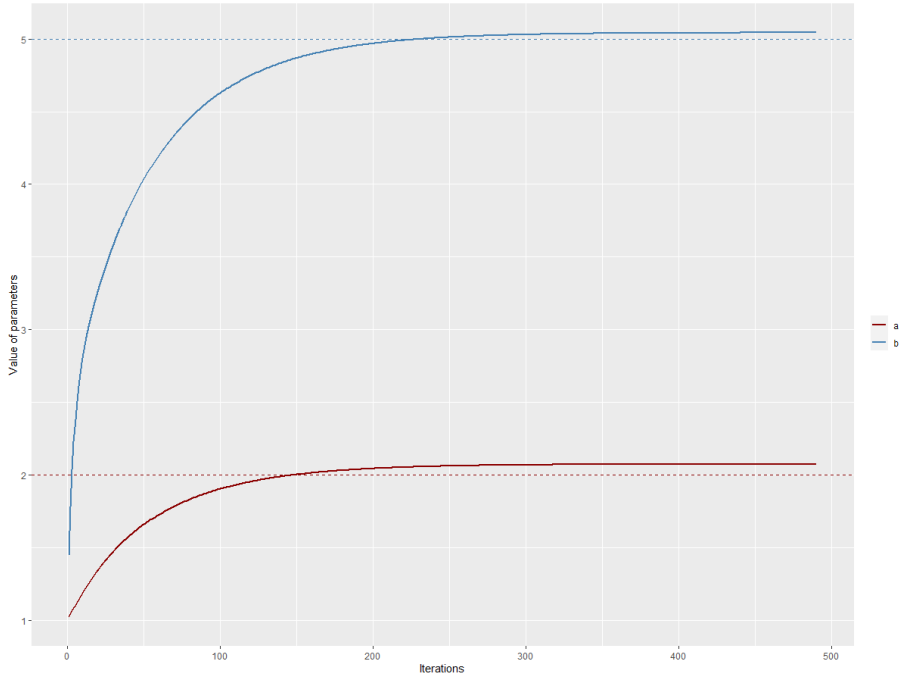
$$\alpha_{new} = \alpha - \frac{\psi(\alpha) - \log\left[\frac{\alpha}{\bar{x}+\alpha_0}\right] - \frac{1}{n}\sum_{k=1}^{n}[\psi(x_k+\alpha_0)]}{\psi_1(\alpha) - \frac{\bar{x}+\alpha_0}{\alpha(1+\beta_0)}}$$

$$\beta_{new} = \frac{\alpha_{new}(1+\beta_0)}{\bar{x}+\alpha_0}$$

Έτσι, το M-step θα μας δώσει τα α_{new} και β_{new}, τα οποία στο επόμενο E-step θα πάρουν την θέση των α_0, β_0, και όταν μετά από διαδοχικά EM-steps επαληθευθεί το κριτήριο της εκφώνησης για δύο διαδοχικές επαναλήψεις, τότε η διαδικασία τερματίζεται. Εφόσον για το NR δεν έχει δοθεί κάποιο κριτήριο τερματισμού, χρησιμοποιούμε αυτό της εκφώνησης. Ο αντίστοιχος κώδικας σε R φαίνεται παρακάτω.

```r
set.seed(10000) #for reproducible results.
x <- rpois(10000,lambda = rgamma(10000,shape=2,rate = 5)) #sample from polya
n <- length(x)
### This metric is our stopping criterion.
metric <- function(a_new,a_old,b_new,b_old){
return(as.numeric((a_new - a_old)^2 + (b_new-b_old)^2))}
### Newton Raphson Function for the Maximization Step
NR <- function(ao,bo,tol){
  a_new <- b_new <- 0
  a <- b <- 1
  repeat{
     a_new <- a - (digamma(a) -log(a/(mean(x)+ao)) -
     (1/n)*sum(digamma(x+ao)))/(trigamma(a)-(mean(x)+ao)/(a*(1+bo)))
    b_new <- (a_new * (1+bo))/(mean(x) + ao)
    if(metric(a_new,a,b_new,b)<=tol){return(c(a_new,b_new))}
    a <- a_new
    b <- b_new}}
### EM step. We start with a,b = 1, find a_new, b_new using Newton Raphson ###
### and then plug them back in and repeat, till the stopping criterion is met. ###
start <- Sys.time()
ao <- bo <- 1
a_new <- b_new <- 0
criterion <- metric(ao,a_new,bo,b_new)
i <- 0
a_s <- b_s <- c()
while(criterion >=1e-10){
  temp <- NR(ao,bo,tol=1e-10)
  a_s[i] <- ao #save values to plot them later
  b_s[i] <- bo # ----//----
  a_new <- temp[1] #new values from NR
  b_new <- temp[2] #----//----
  criterion <- metric(ao,a_new,bo,b_new)
  print(criterion)
  ao <- a_new
  bo <- b_new
  i <- i + 1}
end <- Sys.time()
sprintf('Finished in %d Iterations, and found a=%f and b=%f',i,ao,bo)
print(end-start)
```

Το Output του κώδικα είναι:
“Finished in 491 Iterations, and found a=2.073567 and b=5.043947”
“Time difference of 25.09824 secs ” Όπου με τον παρακάτω κώδικα, παράχθηκε το διάγραμμα 5 που δείχνει την σύγκλιση των α, β στις πραγματικές τους τιμές, συναρτήσει των επαναλήψεων.



Σχήμα 5: Σύγκλιση των α, β στον EM αλγόριθμο.

```r
i_s <- c(seq(1,length(a_s)))
d <- data.table(i_s,a_s,b_s)
gg <- ggplot(data = d,aes(x=i_s)) +
  geom_line(aes(y=a_s,color='darkred'),size=.8)+
  geom_line(aes(y=b_s,color='steelblue'),size=.8) +
  geom_hline(yintercept = 2,color='darkred',linetype='dashed')+
  geom_hline(yintercept = 5,color = 'steelblue',linetype='dashed')+
  scale_color_identity(name = "", breaks = c("darkred", "steelblue"),
                       labels = c("a", "b"), guide = 'legend')+
  xlab('Iterations') + ylab('Value of parameters')
```

Επομένως, βλέπουμε ότι ο αλγόριθμος EM προσεγγίζει πολύ καλά, με ακρίβεια ενός δεκαδικού ψηφίου, τις αρχικές τιμές των παραμέτρων όπως ορίστηκαν στο πρόβλημα, θεωρώντας τες όμως άγνωστες, ενώ χρειάστηκαν 491 επαναλήψεις για να συγκλίνει πλήρως, με βάση το κριτήριο μας.

Όσον αφορά το επαρκές στατιστικό, όπως είδαμε και από την σχέση 3, η σχέση για το Q στο E-step εξαρτάται από τα δεδομένα μόνο μέσω των επαρκών στατιστικών (για log likelihood): $(\sum \log(x), \sum x)$. Το E-step εφαρμόζεται στην $f(\theta \mid x)$ η οποία όπως είδαμε ακολουθεί κατανομή $\text{Gamma}_{(x_k+\alpha_0, 1+\beta_0)}$. Γιαυτό και εμφανίζονται οι μέσες τιμές των επαρκών στατιστικών της, δηλαδή οι όροι: $\mathbb{E}[\log(x)] = \psi(_k+\alpha_0) - \log(1+\beta_0)$ και $\mathbb{E}[x] = \frac{x_k+\alpha_0}{1+\beta_0}$. Αυτό συμβαίνει διότι η κατανομή $f(\theta \mid x)$, όντας Γάμμα, ανήκει στην εκθετική οικογένεια επομένως τα conditional expectations συμπέφτουν με τα επαρκή στατιστικά.

4.

Θεωρήστε το πρόβλημα επιλογής επεξηγηματικών μεταβλητών στην πολλαπλή γραμμική παλινδρόμηση με $n = 50$ παρατηρήσεις και $p = 15$ επεξηγηματικές μεταβλητές. Προσομοιώστε με τη βοήθεια της R (με χρήση της rnorm) τιμές για τις δέκα πρώτες επεξηγηματικές μεταβλητές από την πολυδιάστατη κανονική κατανομή με μέση τιμή $\mathbf{0}$ και πίνακα συνδιακύμανσης τον ταυτοτικό, ενώ για τις υπόλοιπες προσομοιώστε τιμές με βάση τη σχέση:

$$X_{ij} \sim N\left(0.2X_{i1} + 0.4X_{i2} + 0.6X_{i3} + 0.8X_{i4} + 1.1X_{i5}, 1\right), j = 11, \ldots, 15 \text{ και } i = 1, \ldots, 50.$$

Για τη μεταβλητή απόκρισης, προσομοιώστε τιμές, με την βοήθεια της R (με χρήση της rnorm), με βάση τη σχέση:

$$Y_i \sim N\left(4 + 2X_{i1} - X_{i5} + 2.5X_{i7} + 1.5X_{i11} + 0.5X_{i13}, 1.5^2\right), i = 1, \ldots, 50.$$

Εν συνεχεία θεωρήστε το πλήρες πολλαπλό γραμμικό μοντέλο:

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_{15} X_{15} + \varepsilon, \ \varepsilon \sim N\left(0, \sigma^2\right).$$

4.(α') Πολλαπλή Γραμμική Παλινδρόμηση - Επιλογή Μεταβλητών

Εξερευνώντας πλήρως τον χώρο όλων των πιθανών μοντέλων στο πρόβλημα επιλογής επεξηγηματικών μεταβλητών, με τη βοήθεια δικής σας συνάρτησης στην R, βρείτε το μοντέλο εκείνο που ελαχιστοποιεί την τιμή του κριτηρίου BIC.

Αρχικά, πρέπει να κατασκευάσουμε τον πίνακα X με τον τρόπο τον οποίο περιγράφεται στην εκφώνηση, καθώς και την μεταβλητή απόκρισης Y. Στην R εκτελούμε τα εξής:

```
n <- 50
X <- matrix(nrow = n, ncol = 15)
Y <- c()
for (i in 1:10) {#first 10 features.
  X[, i] <- rnorm(n, 0, 1)}
for (j in 11:15) {#rest of the 15 features, as specified.
  for (i in 1:n) {
    X[i, j] <- rnorm(1,mean=0.2 * X[i, 1] + 0.4 * X[i, 2] +
                           0.6 * X[i, 3] + 0.8 * X[i, 4] +
                           1.1 * X[i, 5], sd = 1)}}
# dependent variable Y, as specified.
for(i in 1:n){
  Y[i] <- rnorm(1, 4 + 2 * X[i,1] - X[i,5] + 2.5 * X[i,7] + 1.5 * X[i,11] +
                     0.5 * X[i,13], sd = 1.5)}
```

Στη συνέχεια, πρέπει να υλοποιήσουμε ένα πρόγραμμα το οποίο ελέγχει τον χώρο όλων των πιθανών μοντέλων για το μοντέλο αυτό το οποίο ελαχιστοποιεί την τιμή του κριτηρίου BIC. Το ολόκληρο μοντέλο, έχοντας 15 επεξηγηματικές μεταβλητές, έχει $2^{15} - 1 = 32767$ υποσύνολα. Ένας άλλος τρόπος να το σκεφτούμε είναι ότι τις 15 μεταβλητές μπορούμε να τις διαλέξουμε είτε ανά 1, είτε ανά 2,..., είτε ανά 15, οπότε προκύπτει το άθροισμα $\sum_{i=1}^{15} \binom{15}{i} = 32767$.
Ας ορίσουμε λοιπόν το πρόβλημα. Ένα οποιοδήποτε μοντέλο, μπορεί να γραφεί στην παρακάτω μορφή, όπου $n = 50$ ο αριθμός των παρατηρήσεων, και p ο αριθμός επεξηγηματικών μεταβλητών στο μοντέλο, που ανάλογα το υποσύνολο που θα πάρουμε κινείται από το 1 έως το 15.

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ 1 & x_{31} & x_{32} & \cdots & x_{3p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_n \end{pmatrix}$$

Η λύση αυτού του μοντέλου με τη μέθοδο ελαχίστων τετραγώνων, ανάγεται σε ένα πρόβλημα μεγιστοποίησης:

$$SSE = \|\mathbf{y} - \mathbf{Xb}\|^2 = (\mathbf{y} - \mathbf{Xb})^T(\mathbf{y} - \mathbf{Xb}) = \mathbf{y^T y} - \mathbf{y^T Xb} + \mathbf{b^t X^t Xb}$$

Έτσι, ελαχιστοποιώντας το SSE ως προς το $\mathbf{b}$, έχουμε:

$$\frac{\partial(SSE)}{\partial b} = 0 \iff -\mathbf{y}^T\mathbf{X} + \mathbf{b}^T\mathbf{X^T X} = 0 \iff \hat{\mathbf{b}} = \left(\mathbf{X^T X}\right)^{-1}\mathbf{X^T y}$$

Δεδομένου ότι τα σφάλματα ε ακολουθούν την κανονική κατανομή, η συνάρτηση πυκνότητας πιθανότητας για την εξαρτημένη μεταβλητή Y είναι:

$$f\left(\mathbf{y} \mid \mathbf{x}, \mathbf{b}, \sigma^2\right) = \left(2\pi\sigma^2\right)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{Xb})^{\mathbf{T}}(\mathbf{y} - \mathbf{Xb})\right) = \left(2\pi\sigma^2\right)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2}\hat{\mathbf{e}}^T\hat{\mathbf{e}}\right)$$

όπου όπως είναι προφανές ο πίνακας των υπολοίπων ορίζεται ως: $\hat{\mathbf{e}} = \mathbf{y} - \mathbf{Xb}$. Είμαστε τώρα έτοιμοι να ορίσουμε την λογαριθμική συνάρτηση πιθανοφάνειας:

$$\log(\hat{\mathcal{L}}) = -\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{\hat{\mathbf{e}}^T\hat{\mathbf{e}}}{2\sigma^2}$$

Μεγιστοποιώντας την ως προς σ^2:

$$\frac{\partial \log(\hat{\mathcal{L}})}{\partial \sigma^2} = 0 \Leftrightarrow -\frac{n}{2\sigma^2} + \frac{\hat{e}^T\hat{e}}{2\sigma^4} = 0 \Leftrightarrow \hat{\sigma^2} = \frac{\hat{e}^T\hat{e}}{n}$$

Επομένως, εισάγοντας αυτή την τιμή για το σ^2 στην πιθανοφάνεια, έχουμε την μεγιστοποιημένη πιθανοφάνεια. Τέλος, θα ορίσουμε το BIC, με την τιμή του οποίου θα συγκρίνουμε τα μοντέλα μεταξύ τους.

$$BIC = k \cdot \log(n) - 2 \cdot \log(\hat{\mathcal{L}})$$

όπου $\hat{\mathcal{L}}$ η **μεγιστοποιημένη** συνάρτηση πιθανοφάνειας για το εκάστοτε μοντέλο, $n = 50$ το μέγεθος του δείγματος και k ο αριθμός των παραμέτρων του μοντέλου, που στην πολλαπλή γραμμική παλινδρόμηση συνήθως είναι ο αριθμός των επεξηγηματικών μεταβλητών του μοντέλου +2, ένα για τον σταθερό όρο και ένα για την διασπορά του σφάλματος.
Μια γρήγορη περιγραφή του πως θα υλοποιήσουμε αυτό στην R, αρχικά χρειαζόμαστε μια συνάρτηση που υπολογίζει τους πιθανούς συνδιασμούς, δηλαδή το δυναμοσύνολο των μεταβλητών, αυτό το υλοποιήσαμε με χρήση της συνάρτησης `expand.grid` η οποία επιστρέφει ένα data frame με όλους τους πιθανούς συνδιασμούς των features. Για να γίνει πιο γρήγορο αυτό, το χρησιμοποιούμε ως `expand.grid(rep(list(c(F,T)),d))` ώστε να μας επιστρέψει το δυναμοσύνολο των πιθανών συνδιασμών των μεταβλητών, με Boolean τιμές, True και False.Στην συνέχεια, για κάθε συνδιασμό, κάνουμε fit το πολλαπλό γραμμικό μοντέλο με χρήση της εντολής `lm`, υπολογίζουμε το `BIC` με την βοήθεια της έτοιμης συνάρτησης της R και τελικά κάνουμε append σε ένα data table την τιμή του BIC και το συνδιασμό των μεταβλητών, το οποίο στο τέλος το σορτάρουμε κατάλληλα ώστε να έχουμε το επιθυμητό βέλτιστο μοντέλο. Ο κώδικας μαζί με το summary του τελικού μοντέλου φαίνονται στην **επόμενη σελίδα**.
Το **Output** του κώδικα είναι:
"Time Taken: Time difference of 45.94058 secs"
"Best Model has BIC: 210.347214 and contains variables: 1 5 7 11"
Από το summary παίρνουμε τους συντελεστές για κάθε μεταβλητή και έχουμε την εξίσωση για το τελικό μοντέλο:

$$Y = 4.1873 + 1.8331 \cdot X_1 - 1.3588 \cdot X_5 + 2.0162 \cdot X_7 + 1.9684 \cdot X_{11}$$

Επομένως βλέπουμε πόσο κοντά έχει πλησιάσει και στους συντελεστές και στις μεταβλητές, την εξαρτημένη μεταβλητή Y η οποία έτσι φτιάχτηκε από την αρχή. Παρατηρούμε ότι λείπει μόνο η μεταβλητη X_{13}, η οποία όμως έχει τον πιο μικρό συντελεστή (ίσο με 0.5), το οποίο είναι το δεύτερο μικρότερο BIC. Αυτό συμβαίνει διότι το κριτήριο BIC είναι ανάλογο του αριθμού των παραμέτρων του μοντέλου, επομένως δίνει πιο μεγάλη «ποινή»[1] σε πιο σύνθετα μοντέλα για αυτό και το μοντέλο χωρίς αυτή τη μεταβλητή δίνει μικρότερη τιμή.
Να σημειώσουμε και εδώ, πως μεγάλο ρόλο παίζει και το random seed. Εδώ ξανά το έχουμε σταθερό, ώστε να μπορούμε να επαναλάβουμε τα ίδια αποτελέσματα. Σε δοκιμές να του αλλάξουμε την τιμή, πετύχαμε χαμηλότερο BIC όπου το βέλτιστο μοντέλο είχα όλες τις μεταβλητές από τις οποίες κατασκευάστηκε η Y να εξαρτάται, δηλαδή τις μεταβλητές 1,5,7,11. Όμως αυτό δεν έχει ιδιαίτερο νόημα να το εξετάσουμε περισσότερο, καθώς η τυχαιότητα επηρεάζει το δείγμα άρα και όλη την υπόλοιπη διαδικασία, σημασία έχει να καταλάβουμε για ποιο λόγο το κριτήριο BIC διαλέγει εάν θα κρατήσει μια μεταβλητή, ή όχι.

[1] Ποινή σε εισαγωγικά γιατί απλά αυξάνει η τιμή του, δεν αναφέρεται σε penalty based μεθόδους όπως LASSO, Ridge κ.τ.λ.

```r
set.seed(03400097)# for reproducibility
d <- data.frame(X)
full_glm <- lm(Y ~ .,data = d) # model fit with all variables
get_powerset <- function(set){#function to find subsets
  d <- length(set)
  out <- expand.grid(rep(list(c(F,T)),d))
  out <- as.matrix(out)
  print(out)
  out <- apply(out,1,function(x) set[x])
  return(out)}
start <- Sys.time()
combos <- get_powerset(1:15)#calling the function
bics <- variables <- c()
for(i in 2:length(combos)){#fit every possible subset
  data <- as.data.table(X[,combos[[i]]])
  model <- lm(Y ~., data = data)
  bics[i] <- BIC(model)
  variables[i] <- paste(unlist(combos[[i]]), collapse = ' ')}
end <- Sys.time()
resulting_dt <- na.omit(data.table(BIC=bics, VBs=variables))
resulting_dt <- resulting_dt[order(BIC)]
print("Time Taken:")
end - start
sprintf('Best Model has BIC: %f and contains variables: %s',
        resulting_dt[1]$BIC,resulting_dt[1]$VBs)
final_model <- lm(Y~., data=d[,as.numeric(c(unlist(strsplit(resulting_dt[1]$VBs," "))))])
summary(final_model)
#####OUTPUT OF SUMMARY####
Residuals:
    Min      1Q  Median      3Q     Max
-2.4880 -1.2642 -0.1209  1.1553  3.8496

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.1873     0.2388  17.535  < 2e-16 ***
X1            1.8331     0.2691   6.812 1.94e-08 ***
X5           -1.3588     0.3265  -4.162  0.00014 ***
X7            2.0162     0.3060   6.588 4.17e-08 ***
X11           1.9684     0.1831  10.750 5.15e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 1.653 on 45 degrees of freedom
Multiple R-squared:  0.889,^^IAdjusted R-squared:  0.8792
F-statistic: 90.12 on 4 and 45 DF,  p-value: < 2.2e-16
```

4.(β΄) Μεθοδολογία Lasso

Εφαρμόστε τη μεθοδολογία Lasso με την βοήθεια της βιβλιοθήκης `glmnet` της R και σχολιάστε τα αποτελέσματα. Χρησιμοποιώντας cross-validation σχολιάστε την επιλογή της παραμέτρου ποινής λ καθώς και της παραμέτρου συρρίκνωσης s.

Η μεθοδολογία Lasso, είναι μια τεχνική παλινδρόμησης η οποία επιτυγχάνει ταυτόχρονα επιλογή μεταβλητών αλλά και regularization προς αποφυγή του overfitting. Έτσι, βελτιώνεται η προβλεπτική ικανότητα καθώς και η ακρίβεια του μοντέλου. Ουσιαστικά, προστίθεται ένας επιπλέον όρος στον στην σχέση που ελαχιστοποιούμε κανονικά με τα ελάχιστα τετράγωνα, την οποία καλούμαστε να ελαχιστοποιήσουμε. Δηλαδή ψάχνουμε την λύση του προβλήματος ελαχιστοποίησης:

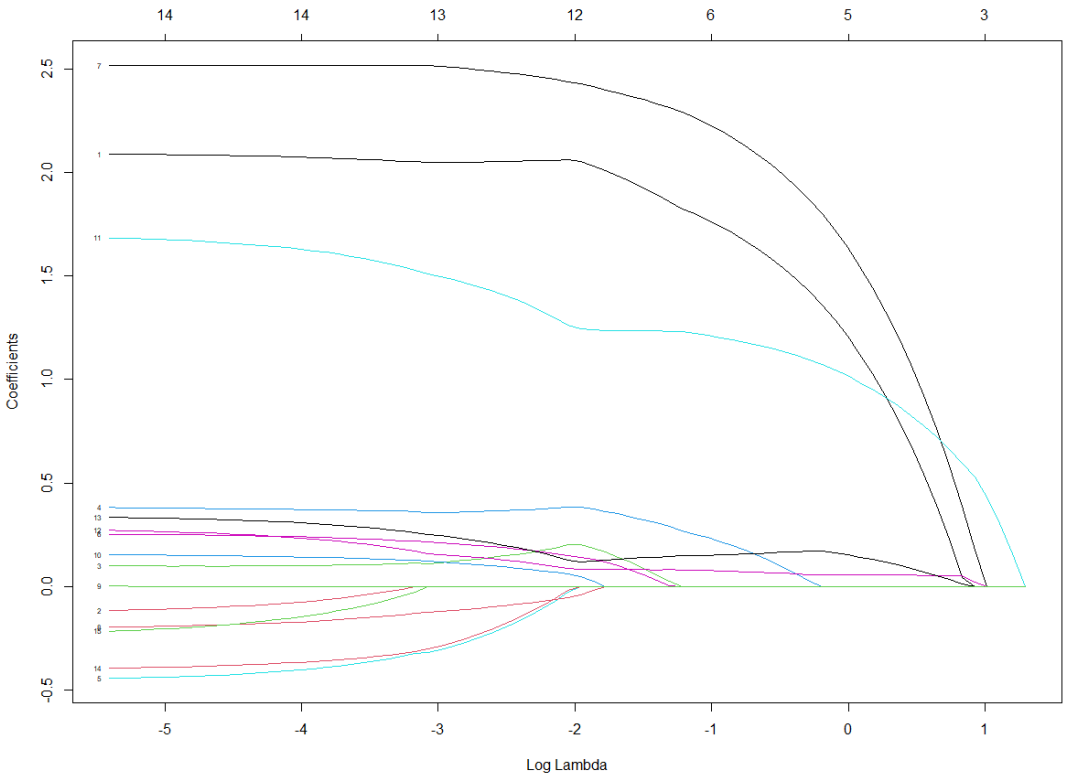
$$\min_{\mathbf{b}} \left\{ (\mathbf{y} - \mathbf{Xb})^T (\mathbf{y} - \mathbf{Xb}) + \lambda \sum_{i=0}^{15} |b_i| \right\}$$

Αυτό λέγεται και L1 regularization καθώς εισάγεται μια ποινή αντίστοιχη με το απόλυτο μέγεθος των παραμέτρων. Η παράμετρος λ καθορίζει το πόσο ισχυρή θα είναι η ποινή. Στην περίπτωση που $\lambda = 0$ τότε δεν υπάρχει καμία ποινή και έχουμε απλή γραμμική παλινδρόμηση. Όσο αυξάνεται η τιμή του, όλο και περισσότερες παράμετροι μηδενίζονται, επομένως στο όριο $\lambda \to \infty$ μηδενίζονται όλες. Άλλο ένα φυσικό επακόλουθο είναι ότι καθώς αυξάνεται το λ, εισάγεται bias στο μοντέλο μας ενώ όσο μειώνεται ανεβαίνει η διασπορά, επομένως πρέπει να βρούμε μια optimal τιμή που να τα εξισορροπεί. Πρωτού προχωρήσουμε στην υλοποίηση, πρέπει να ορίσουμε ένα ακόμα μέγεθος, τον **βαθμό συρρίκνωσης s**. Αυτός ορίζεται ως:

$$s = \frac{|\beta|_1}{\max |\beta|_1}$$

όπου $|\beta|_1$ είναι το άθροισμα των συντελεστών του μοντέλου που μελετάμε, και $\max |\beta|_1$ είναι το μέγιστο άθροισμα των συντελεστών, το οποίο προκύπτει όταν δεν έχουμε καθόλου συρρίκνωση ($\lambda = 0$), δηλαδή είναι το άθροισμα των συντελεστών ελαχίστων τετραγώνων. Με λίγα λόγια, ο συντελεστής s, παίρνει τιμές στο διάστημα $[0, 1]$ και εκφράζει πόσο έχει συρρικνωθεί το μοντέλο μας, όταν είναι 1 δεν έχουμε καθόλου συρρίκνωση, και έχουμε την λύση ελαχίστων τετραγώνων, ενώ όταν είναι 0 τότε όλοι οι συντελεστές β_j είναι μηδενικοί. Δηλαδή: μεγαλύτερο s, μικρότερη συρρίκνωση.
Η υλοποίηση της μεθοδολογίας αυτής θα γίνει με χρήση της βιβλιοθήκης `glmnet` της R. Έχοντας τα X, Y από το (α) ερώτημα, εκτελούμε τα εξής και παίρνουμε το διάγραμμα 6, όπου φαίνονται οι συντελεστές των μεταβλητών συναρτήσει του $\log \lambda$. Με βάση αυτά που είπαμε προηγουμένως, καταλαβαίνουμε ότι όσο αυξάνει το λ (άρα και το $\log \lambda$), τόσο μεγαλύτερο βαθμό συρρίκνωσης (degree of shrinkage) έχουμε.

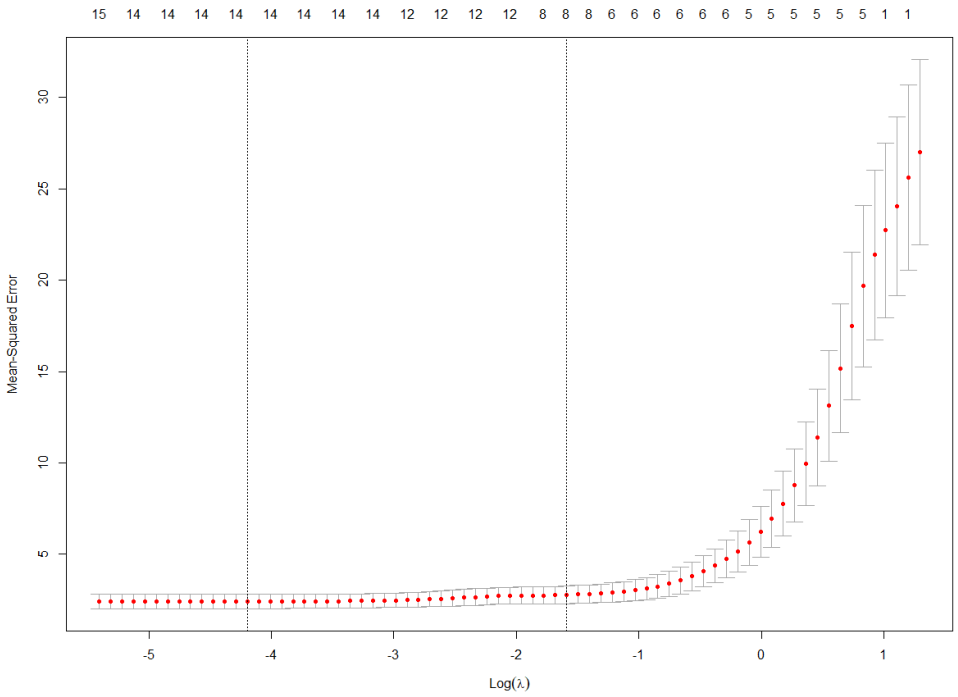
```
1 library(glmnet)
2 fit <- glmnet(X,Y)
3 plot(fit,xvar='lambda',label=TRUE)
```



Σχήμα 6: Lasso Coefficients vs $\log(\lambda)$.

Ο κανόνας εδώ είναι ότι όσο αυξάνει το λ τόσο μηδενίζονται διάφορες μεταβλητές και στο τέλος μένουν οι σημαντικότερες. Γι'αυτό βλέπουμε πως κάποιες από τις μεταβλητές που χρησιμοποιήθηκαν για την δημιουργία της εξαρτημένης μεταβλητής, μηδενίζονται τελευταίες. Επίσης, η μέθοδος Lasso είναι μια μέθοδος η οποία μπορεί να μετριάσει το φαινόμενο της πολυσυγγραμμικότητας, όπου όπως γνωρίζουμε εκ κατασκευής κάποιες από της μεταβλητές X_{ij}, $j = 11, \ldots, 15$ του μοντέλου μας, έχουν οριστεί μέσω γραμμικής σχέσης άλλων μεταβλητών, επομένως αυτές αναμένεται να μηδενιστούν για μικρότερες τιμές του λ, όπως για παράδειγμα η μεταβλητή X_{15}.
Όμως, δεν μπορούμε να κάνουμε επιλογή μεταβλητών «με το μάτι», γι'αυτό και θα εφαρμόσουμε cross validation με σκοπό να βρούμε την ιδανική τιμή του λ καθώς και να έχουμε μια εκτίμηση του προβλεπτικού σφάλματος του μοντέλου. Εκτελούμε τα εξής, και έχουμε το διάγραμμα 7.

```
cv <- cv.glmnet(X,Y)
plot(cv)
#lambda min is the lambda with the minimum cross validation MSE
cv$lambda.min # utputs: 0.01508253
#lambda 1se is the largest value of lambda such that error
is within 1 standard error of the minimum
cv$lambda.1se #outputs: 0.2040739
```



Σχήμα 7: Lasso Mean Squared Error vs $\log(\lambda)$.

Στο διάγραμμα 7 παρατηρούμε δύο κάθετες ευθείες. Κοιτώντας από αριστερά προς τα δεξιά, η πρώτη μας δίνει την τιμή του λ για την οποία ελαχιστοποιείται το Mean Squared Cross Validated Error, η οποία από το output του κώδικα είναι $\lambda_{\min} = 0.015$. Η δεύτερη, μας δίνει την μέγιστη τιμή του λ για την οποία το αναφερθέν σφάλμα βρίσκεται εντός ενός τυπικού σφάλματος, η οποία προκύπτει $\lambda_{1se} = 0.20$. Καθώς $\lambda_{1se} > \lambda_{\min}$, η δεύτερη γραμμή μας δίνει και ένα πιο φειδωλό μοντέλο, το οποίο είναι και το μοντέλο που θα διαλέξουμε. Επομένως, οι συντελεστές είναι οι εξής:

name	coefficient
(Intercept)	4.07
V1	1.96
V3	0.12
V4	0.34
V6	0.08
V7	2.37
V11	1.24
V12	0.08
V13	0.13

Πίνακας 4: λ_{1se} `Model Coefficients`

Το μοντέλο αυτό είναι αρκετά πιο φειδωλό απ'ότι αυτό που θα προέκυπτε με το $\lambda_{\min}$, το οποίο δεν το παρουσιάζουμε, όμως περιέχει όλες τις μεταβλητές εκτός από

μια, δηλαδή 14 από τις 15. Με τον παρακάτω κώδικα, τυπώσαμε τους παραπάνω συντελεστές αλλά και προχωράμε σε υπολογισμό του παράγοντα συρρίκνωσης.

```
coefs <- coef(cv,s = 'lambda.1se')
#multiply with the SDs to standardize variables
zblasso_ <- coefs[-1]*apply(X,2,sd)
zbols_ <- coef(full_glm)[-1]*apply(X,2,sd)
s_lambda.1se <- sum(abs(zblasso_)) / sum(abs(zbols_))#shrinkage factor
s_lambda.1se
```

Τα βήματα που εκτελέθηκαν εδώ, είναι πολλαπλασιασμός των συντελεστών του μοντέλου που βρήκαμε με χρήση της Lasso, με την τυπική απόκλιση, προκειμένου να τυποποιηθούν. Το ίδιο κάνουμε και στο αρχικό πλήρες γραμμικό μοντέλο, το οποίο έχουμε από το (α) ερώτημα. Το ίδιο αποτέλεσμα θα παίρναμε αν θέταμε $\lambda = 0$. Έτσι, ο παράγοντας συρρίκνωσης είναι ο λόγος των αθροισμάτων των συντελεστών των δύο διαφορετικών μοντέλων, ο οποίος προκύπτει **s = 0.63** . Δηλαδή, μπορούμε να πούμε ότι μειώσαμε το αρχικό μας μοντέλο κατά περίπου $\sim 40\%$.

Αναφορές

[1] APA. Efron, B., Tibshirani, R., Tibshirani, R. J. (1994). An introduction to the bootstrap. Chapman Hall/CRC, available at: http://www.ru.ac.bd/stat/wp-content/uploads/sites/25/2019/03/501_02_Efron_Introduction-to-the-Bootstrap.pdf.

[2] Κουρούκλης, Σ., Πετρόπουλος, Κ., Πιπερίγκου, Β., 2015. Θέματα παραμετρικής στατιστικής συμπερασματολογίας. [ηλεκτρ. βιβλ.] Αθήνα:Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Διαθέσιμο στο: https://repository.kallipos.gr/handle/11419/5687.