Stat4DS / Homework 01

Maria Luisa Croci, Valerio Antonini

Excercise 01: Randomize this…

Using RStudio it has been simulated a **one-step randomized algorithm**. The outcomes obtained show that the probability of error approaches to 0 when M (number of iterations) and k (the dimension of the vector z and matrices U, V, W) increase. Choosing at random a $z = (z_1, \ldots, z_k) \in \{0, 1\}^k$ uniformly and constant is the same of choosing each z_i indipendent and uniform from $\{0, 1\}$, infact each of 2^k possibly vector is choosen with probability of $\frac{1}{2^k}$, the ratio of favorable outcomes and total number of possible outcomes.

Repeating the **one-step algorithm** M times for $k = 1$, the probability is:

$$P(Error) = P(UVz = Wz) \leq \left(\frac{1}{2}\right)^M.$$

So the general formula can be written as:

$$P(Error) = P(UVz = Wz) \leq \frac{1}{2^{kM}}.$$

We will have a computational complexity of $O(k^2M)$

```
start_time <- Sys.time()

dim = c(5, 50, 100, 500) #the dimension of z, V, W, U
M = c(100, 1000, 10000) #number of times that the algorithm is repeated

for (i in dim) {
  k = i
  U <- matrix(0, k , k)

  #the three matrices are created at random
  U <- apply(U, c(1,2), function(x) sample(c(0, 1), 1))
  V <- apply(U, c(1,2), function(x) sample(c(0, 1), 1))
  W <- apply(U, c(1,2), function(x) sample(c(0, 1), 1))

  count <- 0

  for (j in M) {

    for (p in 1:j) {

      z <- sample(c(0,1), replace = TRUE, size = k)

      a <- V%*%z
      UVz <- U%*%a
      Wz <- W%*%z

      if(identical(UVz, Wz) == TRUE) { #if the identity is true, increase our counter
        count <- count + 1
      }

    }

    #the probability of error is given by the ratio of the number that the identity is verify and the number of ti
mes M that the algorithm is repeated
    P = count/j
    cat("La probabilita' per k =", k, "e M =", p, "e' pari a", P, "\n")
  }
}
```

```
## La probabilita' per k = 5 e M = 100 e' pari a 0.02
## La probabilita' per k = 5 e M = 1000 e' pari a 0.033
## La probabilita' per k = 5 e M = 10000 e' pari a 0.0364
## La probabilita' per k = 50 e M = 100 e' pari a 0
## La probabilita' per k = 50 e M = 1000 e' pari a 0
## La probabilita' per k = 50 e M = 10000 e' pari a 0
## La probabilita' per k = 100 e M = 100 e' pari a 0
## La probabilita' per k = 100 e M = 1000 e' pari a 0
## La probabilita' per k = 100 e M = 10000 e' pari a 0
## La probabilita' per k = 500 e M = 100 e' pari a 0
## La probabilita' per k = 500 e M = 1000 e' pari a 0
## La probabilita' per k = 500 e M = 10000 e' pari a 0
```

```
#it gives us the speed in fuction of the time
end_time <- Sys.time()
print(end_time - start_time)
```

```
## Time difference of 15.56314 secs
```

We can use the **foreach** package in order to execute faster the algorithm as follows:

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
start_time_foreach <- Sys.time()
registerDoParallel(4)

foreach(dim = c(5, 50, 100, 500)) %dopar% {
  k <- dim

  U <- matrix(0, k , k)

  U <- apply(U, c(1,2), function(x) sample(c(0, 1), 1))
  V <- apply(U, c(1,2), function(x) sample(c(0, 1), 1))
  W <- apply(U, c(1,2), function(x) sample(c(0, 1), 1))

  count <- 0

  foreach(M = c(100, 1000, 10000)) %dopar% {

    for (p in 1:M) {

      z <- sample(c(0,1), replace = TRUE, size = k)

      a <- V%*%z
      UVz <- U%*%a
      Wz <- W%*%z

      if(identical(UVz, Wz) == TRUE) {
        count <- count + 1
      }

    }

    P = count/M
    return(paste("La probabilita' per k =", k, "e M =", p, "e' pari a", P))
  }
}
```

```
## [[1]]
## [[1]][[1]]
## [1] "La probabilita' per k = 5 e M = 100 e' pari a 0.06"
##
## [[1]][[2]]
## [1] "La probabilita' per k = 5 e M = 1000 e' pari a 0.03"
##
## [[1]][[3]]
## [1] "La probabilita' per k = 5 e M = 10000 e' pari a 0.032"
##
##
## [[2]]
## [[2]][[1]]
## [1] "La probabilita' per k = 50 e M = 100 e' pari a 0"
##
## [[2]][[2]]
## [1] "La probabilita' per k = 50 e M = 1000 e' pari a 0"
##
## [[2]][[3]]
## [1] "La probabilita' per k = 50 e M = 10000 e' pari a 0"
##
##
## [[3]]
## [[3]][[1]]
## [1] "La probabilita' per k = 100 e M = 100 e' pari a 0"
##
## [[3]][[2]]
## [1] "La probabilita' per k = 100 e M = 1000 e' pari a 0"
##
## [[3]][[3]]
## [1] "La probabilita' per k = 100 e M = 10000 e' pari a 0"
##
##
## [[4]]
## [[4]][[1]]
## [1] "La probabilita' per k = 500 e M = 100 e' pari a 0"
##
## [[4]][[2]]
## [1] "La probabilita' per k = 500 e M = 1000 e' pari a 0"
##
## [[4]][[3]]
## [1] "La probabilita' per k = 500 e M = 10000 e' pari a 0"
```

```
end_time_foreach <- Sys.time()
print(end_time_foreach - start_time_foreach)
```

```
## Time difference of 13.58106 secs
```

As we can see, this algorithm is more or less 1 second faster than the first without the *foreach* command.

Now, what we have studied is a *p-step* algorithm for $p = 1$.

If we want change to the number of steps to $p = 100$ it is possible to change the bound of the probability of error as:

$$P(Error) = P(UVz = Wz) \leq \left(\frac{1}{2}\right)^p.$$

- Using **Bayes' Theorem** we can calculate $P(E \mid B)$.

Let's define:

– $E = \{$*the matrix identity is correct*$\}$;

– $E^c = \{$*the matrix identity is incorrect*$\}$;

– $B = \{$*the test return that the identity is correct*$\}$.

$$P(E \mid B) = \frac{P(B|E)P(E)}{P(B|E)P(E) + P(B|E^c)P(E^c)}$$

with

– $P(E) = P(E^c) = \frac{1}{2}$;

– $P(B \mid E) = 1$;

– $P(B \mid E^c) \leq \frac{1}{2}$.

So:

$$P(E \mid B) \geq \frac{1 * \frac{1}{2}}{1 * \frac{1}{2} + \frac{1}{2} * \frac{1}{2}} = \frac{2}{3} = 0.666667$$

- Now, repeating the randomized test and having that the matrix identity is correct, we obtain:

– $P(E) \geq \frac{2}{3}$;

– $P(E^c) \leq \frac{1}{3}$;

the probability of *E* in light of *B* changes as:

$$P(E \mid B) \geq \frac{1 * \frac{2}{3}}{1 * \frac{2}{3} + \frac{1}{3} * \frac{1}{2}} = \frac{4}{5} = 0.8$$

- So it's possible to generalize the formula for p-steps as:

– $P(E) \geq \frac{2^p}{2^p+1}$;

– $P(E^c) \leq 1 - \frac{2^p}{2^p+1} = \frac{1}{2^p+1}$;

and applying the Bayes' Theorem:

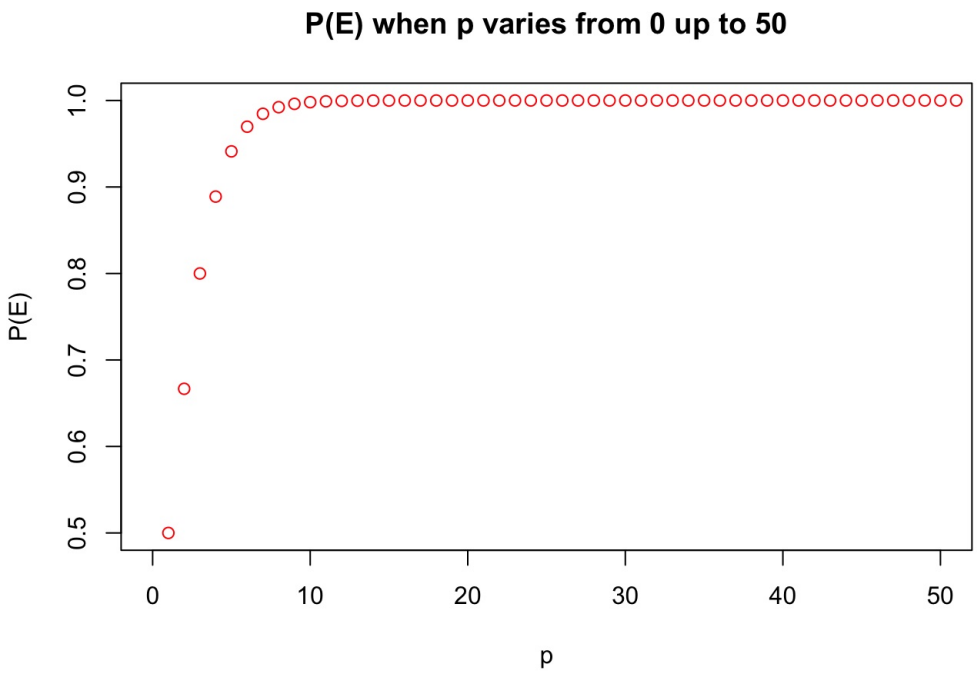
$$P(E \mid B) \geq \frac{1 * \frac{2^p}{2^p+1}}{1 * \frac{2^p}{2^p+1} + \frac{1}{2} * \frac{1}{2^p+1}} = \frac{2^{p+1}}{2^{p+1}+1}$$

- Therefore $\frac{2}{3} \leq P(E) \leq \frac{2^{50}}{2^{50}+1}$ for $1 \leq p \leq 50$

The following plot shows how $P(E)$ varies in function of p:

```
PE <- function(x)(2^x/(2^x+1))

plot(PE(0:50), xlim = c(0,50), ylim = c(0.5,1), main = "P(E) when p varies from 0 up to 50", xlab = "p", ylab = "P
(E)", col = "red")
```



Exercise 02

Version A

Before showing our result, we have just an easy and simple question.

Why pick up your phone and dial its own number?!

Nevertheless, we decided to base our results making a series of research and assuming a priori certain values.

Defined the disjoint events:

– E = {the number gives a busy signal}

– H_1 = {the number is mine}

– H_2 = {the number doesn't exist}

– H_3 = {the number is someone else's}

The question is to calculate the probability that the number is mine given that the number gives a busy signal.

According to *Bayes' Theorem*:

$$P(H_1 \mid E) = \frac{P(H_1) * P(E \mid H_1)}{P(E)} = \frac{P(H_1) * P(E \mid H_1)}{\sum_{i=1}^{3} P(H_i) P(E \mid H_i)}$$

Now, we have deduced and found the following info:

Through ISTAT's source we know that in Italy only the 61,5% of the families have a landline and that in Rome and in its districts live 1.090.012 families. Assuming that the trend of families with a landline is proportionated in all the country, we can obtain how many families own a landline as:

$$61,5\% * 1.090.012 \simeq 670.357$$

Furthermore through Camera di Commercio we obtain the total number of companies in Rome (496.406) and we can also suppose that for each company there is a landline.

So at the end we can estimate that in Rome are available around $670.357 + 496.406 = 1.166.763$ landlines.

So we can define the probabilities as follows:

– $P(E \mid H_1) = 1$ of course calling our number we obtain a busy signal;

– $P(H_1) = \frac{1}{1.166.763}$ the ratio of favorable outcome and the possible outcomes;

– $P(E \mid H_2) = 0$ of course calling a number that doesn't exist, will not return a busy signal but a vocal message;

– We could also estimate $P(H_2)$, but it will be multiplicate to 0, so we don't care about its result;

– Valuating the probability of obtain a busy signal given that the number is somebody else's is not properly easy because it depends by several and differents factors such as the time of the day and the night, the month, how many people live in the same apartment, etc.. For have an easier outcome we assume this probability as $P(E \mid H_3) = \frac{1}{5}$

– $P(H_3) = P(H_1^C) = 1 - P(H_1) = 1 - \frac{1}{1.166.763}$

So, applying Bayes' Theorem we have:

$$P(H_1 \mid E) = \frac{\frac{1}{1.166.763} * 1}{\frac{1}{1.166.763} * 1 + P(H_2) * 0 + (1 - \frac{1}{1.166.763}) * \frac{1}{2}} \simeq 0.000429\%$$

So $\frac{P(H_1 \mid E)}{P(H_1)} \simeq 5$, it means that calling that number and having a busy signal we are 5 times sure that it is our landline.