Università di Roma Tor Vergata
Laurea Magistrale in Informatica

Inferenza Statistica e Teoria dell'Informazione

Scan Statistics

Benedetti Gabriele, Pascale Giulia

Settembre 2022

Abstract

Il seguente documento contiene una breve introduzione agli Scan Statistics (1) per poi passare alla formalizzazione di un problema assegnato nella sezione (2). Successivamente saranno effettuate simulazioni, utilizzando il linguaggio R, nelle sezioni (3) e (4).

Contents

1 Breve introduzione: Scan Statistics **3**

2 Formalizzazione del problema **4**

3 Simulazioni in R **5**
3.1 Caso $a = 0.2, L = 0.8$ 5
3.2 Caso $a = 0.8, L = 0.2$ 6
3.3 Caso $L = 0.5$ 7
3.3.1 Sottocaso $a = 0.5, L = 0.5$ 8
3.3.2 Sottocaso $a = 0.1, L = 0.5$ 8

4 Distribuzione del massimo **9**

5 Conclusioni e considerazioni finali **14**

1 Breve introduzione: Scan Statistics

Gli **Scan Statistics** sono tra i metodi più popolari di **cluster detection**, difatti vengono comunemente utilizzati per verificare se un Point Process[1] sono puramente casuali o se è possibile rilevare eventuali cluster. [2]

Lo studio può proseguire in tre direzioni: dalla scansione spaziale per il rilevamento di cluster, si può variare la finestra di scansione e, infine, il baseline process può essere qualsiasi processo di poisson disomogeneo o processo di bernoulli con cluster di intensità non spiegati dal baseline process.

Questi metodi per statistiche hanno molteplici campi di applicazione, come l'astronomia o la sanità pubblica. Durante la recente emergenza COVID-19 sono stati utilizzati per effettuali analisi statistiche sulla diffisione del virus a Roma e nella regione Lazio [1]. L'obiettivo dello studio in questione fu quello di identificare cluster spaziali di COVID-19 e monitorare l'andamento dei cluster e dei nuovi focolai epidemici nel tempo.

L'**idea** alla base di questi metodi è piuttosto semplice: si tratta di contare gli eventi che ci interessano che si presentano in un arco di tempo o spazio di interesse e di rilevare quando si presentano in numero elevato rispetto ad un numero pre-specificato.

[1] In Statistica e nella teoria della probabilità, un Point Process (anche detto Point Field) è un insieme di punti posizionati casualmente su uno spazio Euclideo o su una linea reale.

2 Formalizzazione del problema

Sia $X_1, ..., X_N$ un campione di v.a. i.i.d. in $[0, 1]$. Come ipotesi nulla H_0 abbiamo che $X_1, ..., X_N \sim U[0, 1]$.
Sia a il punto di partenza della nostra finestra di larghezza L, con $0 < L \leq 1$ e $0 \leq a \leq 1 - L$. Fissato un generico punto X_i, notiamo che la probabilità che X_i appartiene alla nostra finestra è data da

$$P(a \leq X_i \leq a + L) = P(X_i \leq a + L) - P(X_i \leq a) = F(a + L) - F(a) = L$$

Definiamo la v.a. binaria

$$Z_i = \begin{cases} 1 & \text{se } X_i \in [a, a + L] \\ 0 & \text{altrimenti} \end{cases}$$

È evidente che $Z_i \sim \text{Bernoulli}(L)$. Di conseguenza la v.a.

$$Y_{a,L} = \sum_{i=a}^{a+L-1} Z_i$$

che conta il numero di punti presenti nella finestra di lunghezza L, partendo dal punto a, segue una distribuzione binomiale $Y_{a,L} \sim \text{Bin}(N, L)$ (sempre sotto H_0).

Osservare che il numero di punti all'interno della finestra non dipende dalla posizione di quest'ultima ma dalla dimensione della stessa.

Intuitivamente è facile verificare quest'ultima affermazione in quanto, sotto H_0, non ci sono punti di accumulo per via del fatto che questi sono distribuiti uniformemente a caso in $[0, 1]$.

La distribuzione di $Y_{a,L}$ risulta quindi essere

$$P(Y_{a,L} = k) = \binom{N}{k} L^k (1 - L)^{N-k}$$

con media $\mu = NL$ e varianza $\sigma^2 = NL(1 - L)$.

3 Simulazioni in R

In questa sezione saranno mostrate diverse simulazioni svolte tramite il linguaggio di programmazione R, utilizzato principalmente per calcoli statistici.

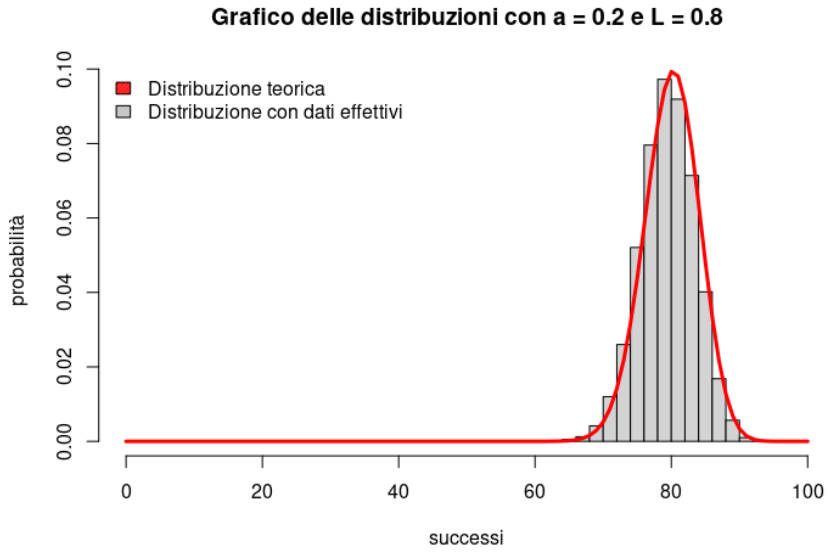
Per evitare inutili ripetizioni, in questo documento sarà inserito solo il codice relativo al caso specifico $a = 0.2, L = 0.8$. Nelle altre simulazioni è stato utilizzato lo stesso codice con le opportune modifiche.

```r
N = 100
a = 0.2
L = 0.8
n = 6000

test <- function(N, a, L){
    X <- runif(N)
    Z <- (X >= a) & (X <= a + L)
    Y <- sum(Z)
    return(Y)
}

Y <- replicate(n, test(N,a,L))
print(Y)
hist(Y, main = "Grafico delle distribuzioni con a = 0.2 e L = 0.8", xlab = "successi",
    ylab = "probabilità", freq=FALSE, xlim = c(0,100))
lines(0:N, dbinom(0:N, size=N, prob=L), col='red', lw=3)
legend("topright", c("distribuzione teorica", "distribuzione con dati effettivi"),
    bty = "n", density = c(100, 100),
    fill=c("red", "gray"))
```

Implementazione in R

3.1 Caso $a = 0.2, L = 0.8$

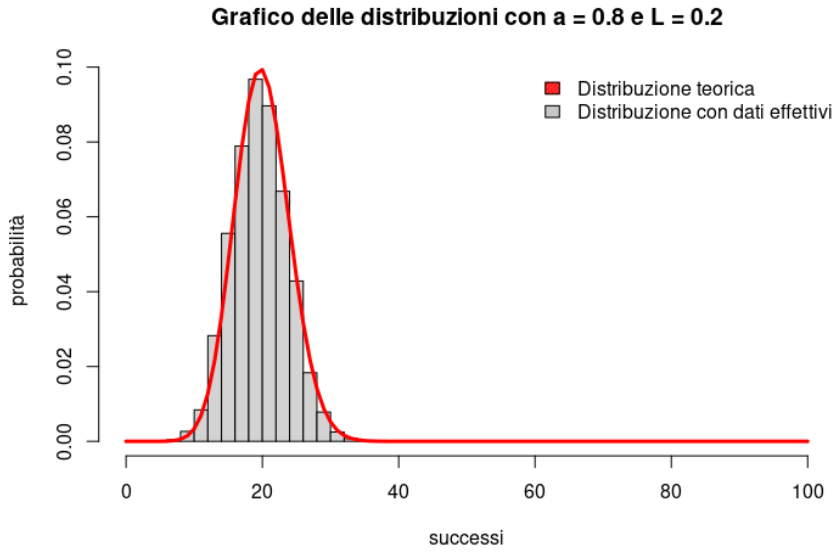
Una prima simulazione è stata effettuata ponendo la larghezza della finestra $L = 0.8$. Di conseguenza il valore massimo di a è 0.2 ($0 \leq a \leq 0.2$). La simulazione ha prodotto il seguente grafico



È possibile notare che la distribuzione ottenuta dai dati effettivi (cioè con gli N punti generati in modo randomico dalla funzione `runif()`) segue la distribuzione teorica.

3.2 Caso $a = 0.8, L = 0.2$

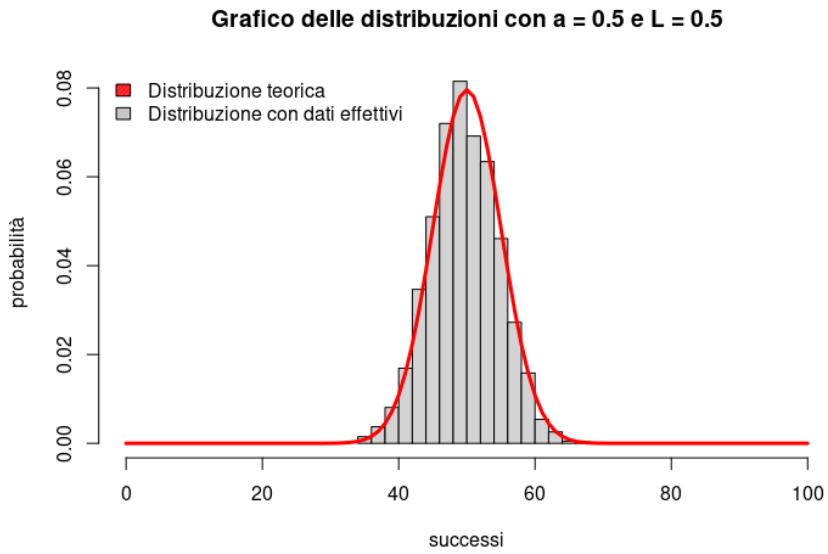
Si è deciso di testare i valori opposti, cioè di voler effettuare la simulazione su una finestra di lunghezza ristretta $L = 0.2$ e di scegliere un valore di $a = 0.8$ ($0 \leq a \leq 0.8$). In questo caso la simulazione ha prodotto il grafico:



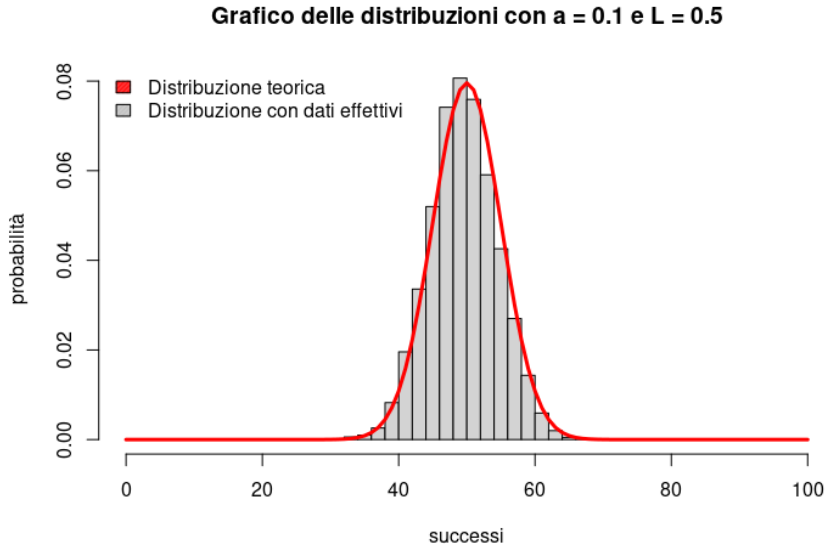
3.3 Caso $L = 0.5$

A questo punto si è deciso di fissare la finestra $L = 0.5$ e di modificare solamente il valore di a, per verificare se la percentuale del numero di successi poteva essere influenzata dal punto di partenza della finestra.

3.3.1 Sottocaso $a = 0.5, L = 0.5$



3.3.2 Sottocaso $a = 0.1, L = 0.5$



4 Distribuzione del massimo

A questo punto facciamo scorrere la finestra di lunghezza L (per comodità fisseremo L=0.1, ovviamente i risultati sono gli stessi con altri valori di L) sul segmento [0,1] e salviamo la concentrazione massima di punti in una v.a. che inseriamo in un vettore M. Iteriamo il procedimento generando ogni volta i punti nel segmento e inseriamo il valore massimo in M. Di seguito il codice:

```
Y_max <- function(N, L, delta=0.001) {
    temp <- c(0.0, 0.0)
    X <- runif(N)
    for(a in seq(from = 0.0, to = 1-L, by = delta)) {

        Z <- (X >= a) & (X <= a+L)
        Y = sum(Z)
        if (Y > temp[1]){
            temp[1] <- Y
            temp[2] <- a
        }
    }
    return(temp)
}
#itero la funzione Y_max e salvo tutti i valori massimi trovati in un vettore M
maxima <- function(N, L, times=1000) {
    M <- c()
    for(i in 1:times){
        M[i] <- Y_max(N,L)[1]
    }
    return(M)
}

M <- maxima(N, L)
hist(M, main = "Grafico della distribuzione di M con N = 7000")
```

Ora vogliamo analizzare la distribuzione di M e la sua media. Dalle simulazioni si ottengono i seguenti grafici delle distribuzioni:

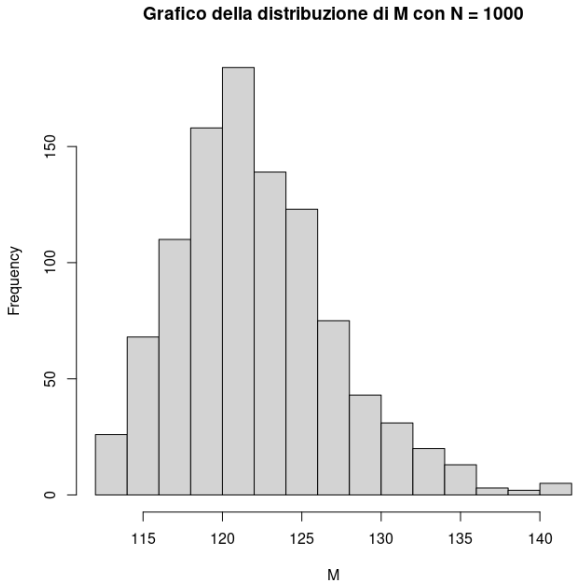


Grafico della distribuzione di M con N = 1000
Frequency
0
50
100
150
115
120
125
130
135
140
M

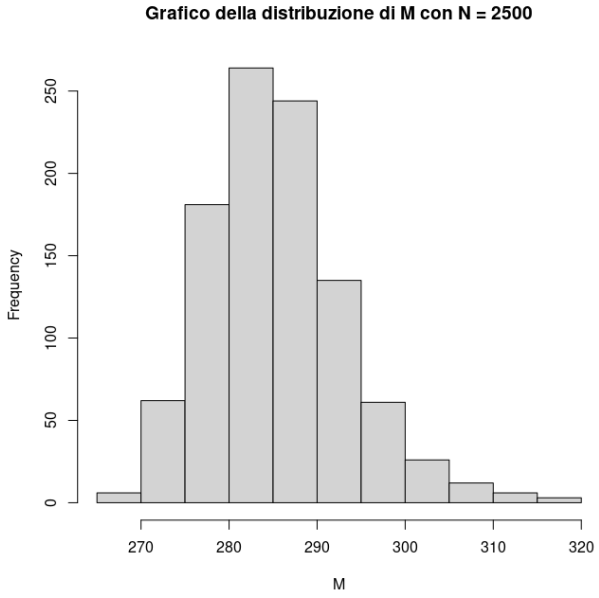
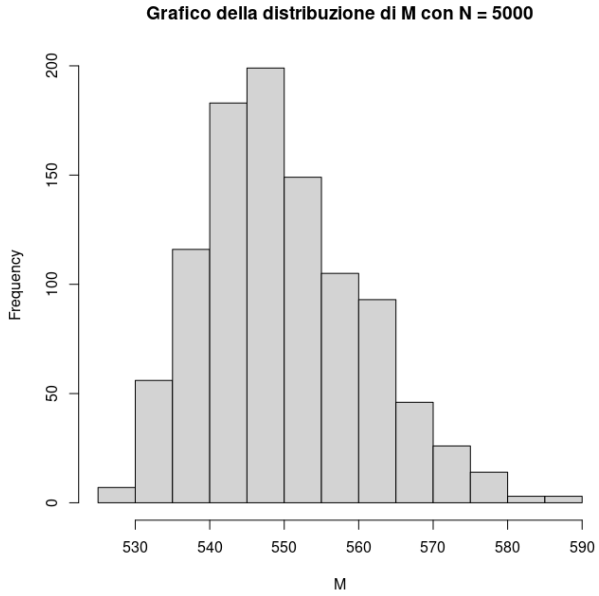
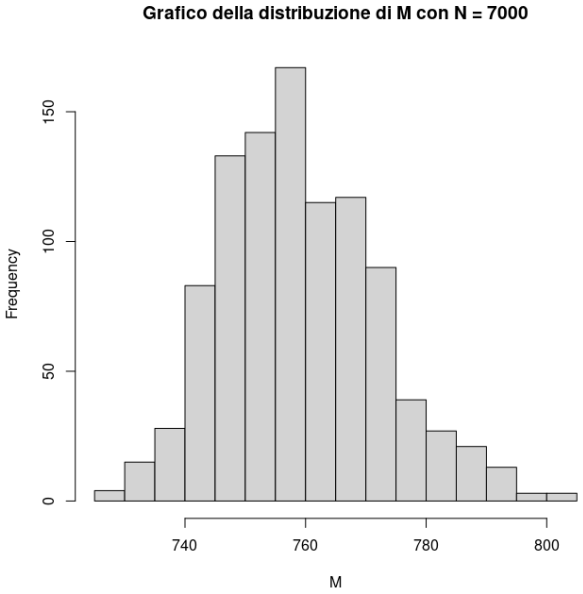


Grafico della distribuzione di M con N = 2500
Frequency
0
50
100
150
200
250
270
280
290
300
310
320
M





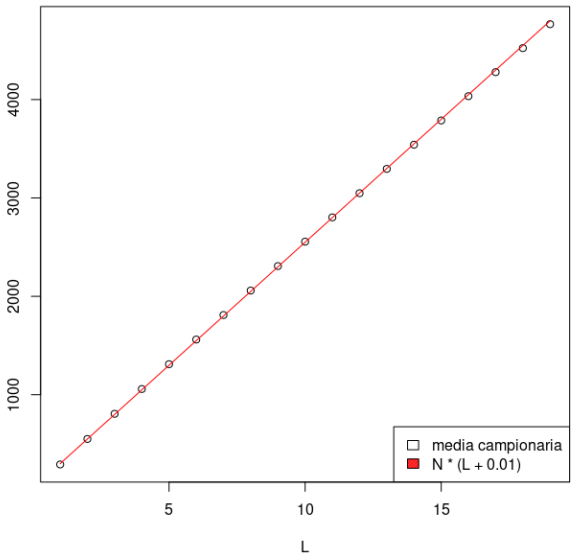
In base a questi istogrammi, sembrerebbe che la media sia leggermente spostata rispetto ad una binomiale di un fattore additivo rispetto ad L. Dopo diverse simulazioni possiamo *empiricamente* approssimare la media della distribuzione con $N(L + 0.01)$.

Nella seguente tabella mostriamo alcuni dei risultati ottenuti al variare di L.

L	$\overline{X}$	$N(L+0.01)$
0.050	291.258	300.000
0.100	550.456	550.000
0.150	806.108	800.000
0.200	1058.257	1050.000
0.250	1310.473	1300.000
0.300	1560.921	1550.000
0.350	1810.092	1800.000
0.400	2058.665	2050.000
0.45	2307.68	2300.00
0.50	2555.91	2550.00
0.550	2802.666	2800.000
0.600	3048.254	3050.000
0.650	3295.494	3300.000
0.700	3541.377	3550.000
0.750	3787.806	3800.000
0.800	4034.315	4050.000
0.850	4278.968	4300.000
0.900	4523.118	4550.000
0.950	4766.066	4800.000

Table 1: Campione su $N = 5000$.

Osservando l'andamento della media campionaria e del valore ipotizzato $N(L + 0.01)$ si nota che il grafico è simile (salvo minime fluttuazioni).



Il codice utilizzato per ottenere la precedente tabella e l'ultimo grafico è il seguente:

```
times = 1000
windows <- c()
means <- c()
expected <- c()

for(l in seq(from = 0.05, to = 0.95, by = 0.05)) {
    M <- maxima(N, l, times)
    # hist(M, main = "grafico della distribuzione del massimo con N = 5000")
    print(c(l, sum(M)/times, N*(l + 0.01)))

    windows <- append(windows, l)
    means <- append(means, sum(M)/times)
    expected <- append(expected, N*(l + 0.01))
}
plot(means, xlab="L")
lines(expected, col='red')
legend("bottomright", c("media campionaria", "N * (L + 0.01)"), density=c(100,100), fill=c("white", "red"))
```

5 Conclusioni e considerazioni finali

Possiamo concludere questo documento mettendo in evidenza che, come ipotizzato nella sezione 2, la percentuale di successi dipende solo dalla larghezza L e dagli N punti. A supporto di questa conclusione ci sono le simulazioni effettuate, in particolare i test in cui è stato fissato L ed è stato variato solo il valore di a in 3.3.1 e 3.3.2.
Inoltre è stata verificata l'ipotesi che la v.a. Y (vista in 2), che conta il numero di successi in N punti totali, segue effettivamente una distribuzione binomiale. Abbiamo successivamente analizzato nella sezione 4 la distribuzione del massimo notando che è simile ad una binomiale ma con una media che ha un fattore additivo di circa 0.01 rispetto alla media di una binomiale.

References

[1] Chiara Badaloni, Federica Asta, Paola Michelozzi, Francesca Mataloni, Enrico Di Rosa, Paola Scognamiglio, Francesco Vairo, Marina Davoli, Michela Leone, *"Analisi spaziale per l'identificazione di cluster di casi durante l'emergenza COVID-19 a Roma e nel Lazio"*, 2020.
Articolo disponibile qui.

[2] Martin Kulldorff, *A Spartial Scan Statistic*, 1997.

[3] Jie Chen, Joseph Glaz, *Approximations and Inequalities for the Distribution of a Scan Statistic for 0-1 Bernoulli Trials*, 1996.

[4] Joseph Glaz, *Approximations and Bounds for the Distribution of the Scan Statistic*, Journal of the American Statistical Association Vol. 84, No. 406 (Jun., 1989), pp. 560-566 (7 pages)

[5] Neville Nagarwalla, *A Scan Statistic with a variable window*, 1996.