Laboratorio di Statistica Computazionale - 1

Matteo Grigoletto[1]

[1] Il materiale presentato è basato sugli appunti di Domenico De Stefano.

Argomenti di oggi

- Metodi di simulazione diretti.
  - Generazione mediante inversione.
  - Generazione mediante trasformazione.
- Metodi di simulazione indiretti.
  - Generazione mediante accettazione/rifiuto (A/R).

Alcuni esercizi per oggi

- Esercizio n. 1
  - **a.** Utilizzare il metodo della simulazione per inversione per simulare da una v.c. esponenziale.
  - **b.** Verificare l'adattamento su $n = 10000$ valori simulati.
- Esercizio n. 2
  - **a.** Simulare mediante accettazione/rifiuto da una Beta$(2, 2)$ sull'intervallo (0,1) e verificare l'adattamento e la proporzione di accettazione.
  - **b.** Si provi a cambiare il parametro k e si commenti il risultato.
- Esercizio n. 3
  - **a.** Simulare una normale standard troncata ai soli valori positivi, usando un algoritmo A/R basato su una distribuzione esponenziale.
- Esercizio n. 4
  - **a.** Si generi una normale standard mediante trasformazione (Box-Muller e trasformazione polare).

Algoritmo base/ 1

Algoritmo di Wichmann-Hill (1982). Numeri pseudo-casuali uniformi in [0,1].

```
.current.seed <- c(123,456,789) # Seme iniziale
runif.wh <- function(n){
   a <- c(171,172,170)
   b <- c(30269,30307,30323)
   s <- .current.seed
   u <- rep(0, n)
   for(i in 1:n){
     s <- (a*s) %% b # Aggiornamento sequenza (n volte)
     u[i] <- sum(s/b) %% 1 # Parte frazionaria della somma
                           # della sequenza dei numeri in s
   }
   .current.seed <<- s # Seme finale
   u
}
```

Note esercizio 1

- Il più semplice metodo (diretto) di simulazione è quello detto di inversione e si basa sul seguente risultato: se Y è un v.c. con FdR F_Y, allora $U = F_Y(Y) \sim U(0,1)$. Dunque, se $U \sim U(0,1)$ si ha che:

$$Y = F_Y^{-1}(U) \text{ ha funzione di ripartizione di } F_Y.$$

- Utilizzabile quando la funzione quantile è esprimibile in forma chiusa.
- Per l'es. 1 si parta da $Y \sim \mathrm{Exp}(\lambda)$, ossia:

$$F_Y(y) = 1 - e^{-\lambda y}, \; y \geq 0.$$

- Da questa funzione si ricavi $F_Y^{-1}(U)$ (e per generare da U si usi la funzione creata in precedenza, ossia `runif.wh`).

Funzione A/R

```
r.accetto.rifiuto <- function(n, f, g, rg, k, report=TRUE){
   y <- double(n)
   ntry <- 0
   for (i in 1:n){
     done <- FALSE
     while (!done){
       ntry <- ntry + 1
       z <- rg(1)
       u <- k * g(z) * runif.wh(1)
       if (u <= f(z)) done <- TRUE
     }
     y[i] <- z
   }
   if (report)
     cat("r.accetto.rifiuto:  ho fatto",ntry,"tentativi\n")
   y
}
```

Note Esercizio 2

- Con riferimento all'algoritmo scritto in precedenza, quali funzioni `g` e `rg` utilizzereste?
- Si trovi il massimo della $\text{Beta}(2, 2)$ nell'intervallo di interesse.
- A tale scopo si può usare R (valutando la densità della Beta nell'intervallo di interesse).
- Perché ci serve questo valore? Quale argomento della funzione rappresenta?
- Per la parte **b.** dell'esercizio usare $k = 1, 2, 3$.

Note Esercizio 3

- Si consideri la funzione di densità della normale standard troncata ai soli valori positivi:

$$f(y) = \frac{2}{\sqrt{2\pi}}\, e^{-y^2/2}, \text{ con } y > 0.$$

- Usando una densità esponenziale $g(y) = e^{-y}$, si determini la costante k richiesta.
- Per generare da una distribuzione esponenziale si usi la funzione `rExp` creata nell'esercizio n. 1.

Note Esercizio 4

- L'algoritmo di generazione per trasformazione secondo Box-Muller consente di simulare numeri pseudo-casuali da una normale standard.
- L'algoritmo è il seguente.
  - Si considerano due variabili casuali indipendenti U(0,1): U_1 e U_2;
  - si pone $R = \sqrt{-2\,log(U_1)}$ e $\theta = 2\,\pi\,U_2$;
  - si pone $X = R\,cos(\theta)$ e $Y = R\,sin(\theta)$;
  - X e Y sono due variabili casuali normali standard indipendenti.
- Si scriva l'algoritmo in R e si verifichino i risultati usando come confronto la funzione `dnorm`, inclusa in R.
- Le funzioni `log`, `sin` e `cos` non sono sempre veloci da calcolare, ma spesso questo algoritmo viene preferito ad altri per la sua semplicità (che compensa l'eventuale leggera perdita di efficienza).
- Come nota a margine aggiungiamo che, anche se può sembrare strano, è possibile generare da una normale anche usando il metodo per inversione. Infatti, non esiste una forma esplicita per l'inversa della funzione di ripartizione della distribuzione normale, ma ne esistono versioni approssimate.