Gianfranco Cavedon / Roberto Soccorsi

Teoria e algoritmi per la generazione di variabili casuali univariate

www.aracne-editrice.it
info@aracne-editrice.it

via Raffaele Garofalo, 133 A/B
00173 Roma
redazione: (06) 72672222 – telefax 72672233
amministrazione: (06) 93781065

ISBN 88-7999-808-0

I edizione: ottobre 2004

INDICE

- PREFAZIONE -

Lo studio ha una impostazione teorico-pratica: accanto ai problemi di metodologia che si sono dovuti risolvere per garantire la casualità e diminuire la ciclicità dei numeri pseudocasuali, esistono, infatti, altri problemi di carattere tecnico per la realizzazione pratica degli algoritmi.

Due sono gli obiettivi di questo lavoro: descrizione analitica degli algoritmi per la generazione di v.c. continue e discrete e realizzazione di un "package" monotematico per la generazione di specifiche variabili casuali Univariate, sia Discrete sia Continue, in modo che il ricercatore o lo studente possa disporre di uno strumento svincolato da qualsiasi contesto di programmazione e soprattutto strutturato per la generazione efficiente, dal punto di vista del tempo, della precisione e dell'accuratezza, di un numero molto elevato di v.c.

Il secondo obiettivo è stato realizzato in quattro fasi successive, ciascuna delle quali ha richiesto tempi significativi per lo studio, l'analisi, la valutazione, la programmazione e la messa a punto dei diversi algoritmi che sono stati inseriti nel progetto definitivo del software.

Esaminiamo più dettagliatamente ciascuna delle quattro fasi mettendo in evidenza i momenti critici di ciascuna di esse.

Fase 1: Sono stati attentamente studiati i metodi più noti di generazione di v.c. Uniformi, evidenziando, per ciascuno di essi, vantaggi e svantaggi che potevano presentarsi al momento della loro utilizzazione. Sono inoltre state ideate alcune varianti, al fine di ridurre il livello di autocorrelazione o di dipendenza insito nel metodo stesso. La realizzazione pratica degli algoritmi è stata particolarmente studiata in modo tale che questi risultassero quanto più possibile ottimizzati, soprattutto nell'ottica di un loro utilizzo su personal computer, mantenendo però come obiettivo primario quello della precisione e dell'accuratezza.

Fase 2: Sono stati analizzati alcuni metodi di trasformazione delle v.c. Uniformi, adottati sia in diverse librerie sia in linguaggi di programmazione di IV generazione - IMSL, NAg, SAS, SPSS, S-PLUS - per ottenere distribuzioni di interesse generale nei problemi di simulazione, sempre nell'ottica di un loro utilizzo su pc.

L'analisi è stata condotta consultando numerose pubblicazioni specialistiche, tutte citate in bibliografia. Nel corso di questa fase sono state evidenziate delle imprecisioni nella descrizione di alcuni algoritmi.

Fase 3: Gli algoritmi individuati sono stati programmati tutti in doppia precisione e dopo una serie di prove si è passati dalla fase prototipale a quella definitiva.

Fase 4: Rappresenta la fase di valutazione e di scelta definitiva dei diversi algoritmi da implementare. Infatti, è stato possibile procedere alla selezione dell'algoritmo *ottimo* in base: a) al valore di alcuni parametri delle distribuzioni - media, moda, varianza, ecc. - ottenuti sperimentalmente a partire da diverse combinazioni di valori dei parametri iniziali tipici di una specifica distribuzione; b) ai tempi di elaborazione; c) al numero di istruzioni necessarie per la programmazione dell'algoritmo; d) alla lunghezza del codice oggetto generato.

Nella scelta dell'algoritmo è stato quindi privilegiato quello che ha soddisfatto il maggior numero di questi requisiti. A volte, tuttavia, algoritmi più accurati e precisi non sono necessariamente anche più veloci e più brevi; si è dunque preferito optare per algoritmi più

lenti ma più affidabili dal punto di vista della precisione e dell'accuratezza. I parametri delle distribuzioni sono stati ottenuti con il programma **CLASSIG**, contenuto nel CD-ROM insieme al software **GEVARA**, che è il risultato di un intenso lavoro di analisi, comparazione e valutazione effettuato per 21 diversi generatori di v.c.

Si è, infine, proceduto a un'ulteriore verifica della qualità delle v.c. pseudocasuali generate, valutando le rappresentazioni grafiche ottenute mediante il programma **GNUPLOT** dopo avere opportunamente discretizzato le v.c. continue e ottenuto la distribuzione di frequenza con il programma **CLASSIG** .

In conclusione, riteniamo che il lavoro realizzato, per cui è stato necessario un notevole impegno a livello teorico e pratico, potrà essere utile a ricercatori e studenti che abbiano la necessità di effettuare lavori sia teorici sia applicativi nel campo della simulazione.

La facilità di utilizzo del "package" permetterà, infatti, di focalizzare direttamente la loro attenzione sul problema oggetto di studio, senza doversi preoccupare di conoscere specifici linguaggi di programmazione per poter utilizzare la *procedura* necessaria a generare un certo numero di determinazioni di una particolare v.c.

Al fine di consentire una migliore utilizzazione del software, il lavoro è stato corredato da un testo in cui, oltre a una trattazione di carattere generale sul tema della generazione automatica di v.c., viene anche presentata una descrizione schematica di ogni algoritmo considerato.

CAPITOLO I
GENERATORI DI DI SEQUENZE DI NUMERI PSEUDOCASUALI

Par. 1.1 PREMESSA

La necessità di generare numeri pseudocasuali, ottenuti all'inizio in modo manuale o meccanico [Knuth, 1981, vol II, pag. 2] e successivamente in modo automatico mediante un elaboratore, deriva da diverse esigenze sia di carattere teorico-scientifico, sia di carattere empirico tra cui:

- Soluzione di problemi mediante simulazione di sistemi a eventi discreti o continui.
- Campionamento per lo studio di problemi appartenenti a diverse branche della ricerca scientifica.
- Generazione di dati per il test di Algoritmi numerici.
- Metodo di Montecarlo per lo studio e la simulazione dei più disparati problemi matematici, statistici, o anche di carattere ludico (giochi di carte, lancio di dadi, ecc.).
- Metodi statistici ad alta intensità computazionale: Cross Validation, Bootstrap, ecc..

Le formule matematiche utilizzate per generare in modo automatico le sequenze di numeri, ciascuno dei quali dipende dai precedenti, non garantiscono una perfetta casualità, ma questo limite può essere parzialmente corretto scegliendo il periodo della successione quanto più ampio possibile e sottoponendo la sequenza generata ad alcuni test di casualità.

Procedendo in questo modo si genera una successione di numeri sia pure con periodo lungo, ma tuttavia finito. Per questo motivo si può parlare di sequenze pseudocasuali e non casuali.

Il primo metodo aritmetico per generare numeri pseudocasuali mediante un elaboratore – proposto da von Neumann e Metropolis nel 1946 – è conosciuto come metodo del "centro del quadrato", in cui ogni numero della sequenza è ottenuto prendendo la cifra centrale del quadrato del numero precedente. Detto metodo fu poi abbandonato a favore di quello delle "Congruenze", perché lento e statisticamente insoddisfacente.

I processi di generazione di numeri pseudocasuali non sono tutti accettabili dal punto di vista della casualità e dei requisiti richiesti affinché le sequenze generate possano definirsi pseudocasuali. I requisiti sono sinteticamente i seguenti:

1. I numeri generati devono essere uniformemente distribuiti e statisticamente indipendenti nella sequenza.
2. La sequenza generata deve essere riproducibile.
3. La sequenza deve poter avere un periodo di lunghezza arbitraria e più ampio possibile.

Un buon metodo di generazione [Fishman, 1996, pag. 587] deve essere quanto più possibile efficiente in termini di tempo di elaborazione e, nel caso di elaborazione automatica, impegnare al minimo la memoria centrale.

Questo lavoro si occupa dello studio di diversi metodi di generazione di sequenze di numeri pseudocasuali di v.c. assolutamente continue: Beta, Gamma, Normale, ecc., e discrete: Binomiale, Ipergeometrica, Poisson, ecc. A tale scopo si utilizza, quando è possibile, la relazione che esiste tra la funzione di ripartizione di dette v.c. e la v.c. Uniforme continua $U(0,1)$ - Metodo dell'Inversione analitica – altrimenti si ricorre ad altri metodi – quello di Accettazione e Rifiuto (**A/R)** in diverse versioni, per esempio – che utilizzano sempre la v.c. U.

Questo aspetto sarà trattato diffusamente nel III Capitolo.

Il processo per la generazione della v.c. U è così articolato:

- Si genera un certo numero di valori discreti interi $Z_i \in [0, m-1]$, in cui m, che rappresenta l'ampiezza massima del periodo della successione delle Z_i, dipende dall'estensione di una voce intera del sistema di elaborazione utilizzato, in genere $m=2^{31}-1$.

Detti valori vengono generati con metodi iterativi basati sulla relazione congruenziale:

$$Z_i \equiv f\left(Z_{i-1}; \underline{A}\right) \bmod m \quad i \geq 1,$$

oppure in modo equivalente, come si vedrà nel paragrafo 1.3.2:

$$Z_i = f\left(Z_{i-1}; \underline{A}\right) \bmod m \quad i \geq 1,$$

dove f è una funzione lineare o non lineare e $\underline{A}$ è l'insieme dei parametri che la caratterizzano.

La generazione dei valori avviene in modo non casuale, ma mediante un processo deterministico dal momento che ciascuno dei valori generati dipende da quello precedente:

$$Z_1 = f\left(Z_0; \underline{A}\right) \bmod m$$
$$Z_2 = f\left(Z_1; \underline{A}\right) \bmod m = f\left(f\left(Z_0; \underline{A}\right) \bmod m\right) \bmod m$$
$$\ldots\ldots\ldots\ldots$$
$$\ldots\ldots\ldots\ldots$$

- Successivamente, le Z_i vengono normalizzate rispetto a m ottenendo:

$$Y_i = Z_i / m; \quad Y_i \in (0, \frac{m-1}{m}) \quad i \geq 1.$$

Si pone quindi il problema di generare valori Z_i in modo tale che il periodo m sia quanto più ampio possibile affinché i valori di Y_i possano essere assimilabili a determinazioni di una v.c. Uniforme continua $U(0, 1)$.

La determinazione delle $\{Z_i\}$, per particolari valori dell'insieme dei parametri $\underline{A}$ e del modulo m, come vedremo in seguito, possono essere considerate come determinazioni di una v.c. Uniforme discreta nell'intervallo $[0 - m-1]$ e a valori interi:

$$\Pr\{Z = i\} = \frac{1}{m} \quad i = 0, 1, \ldots\ldots, m-1,$$

il cui valore atteso e la varianza sono:

$$E(Z) = \frac{m-1}{2}$$
$$Var(Z) = \frac{(m-1)^2}{12};$$

pertanto anche le Y_i possono essere considerate, a loro volta, come determinazioni di una v.c.

discreta Uniforme $U\left(0, \frac{m-1}{m}\right)$.

Il valore atteso e la varianza della v.c. *Y* e della v.c. Uniforme continua *U(0,1)* sono rispettivamente:

$$\begin{cases} E(Y) = \frac{1}{2}\frac{m-1}{m} & ;\ E(U) = \frac{1}{2} \\ Var(Y) = \frac{1}{12}\frac{(m-1)^2}{m^2} & ;\ Var(U) = \frac{1}{12}. \end{cases}$$

Se *m* tende all'infinito, i momenti delle due v.c. coincidono, per cui si può concludere che la v.c. *Y* è una v.c. Uniforme continua *U*(*0, 1*) [Kennedy e Gentle, 1980, pag. 142].

Dei numerosi metodi per la generazione di successioni di numeri pseudocasuali proposti da diversi autori, abbiamo considerato esclusivamente quelli che in letteratura sono più noti e quindi maggiormente utilizzati.

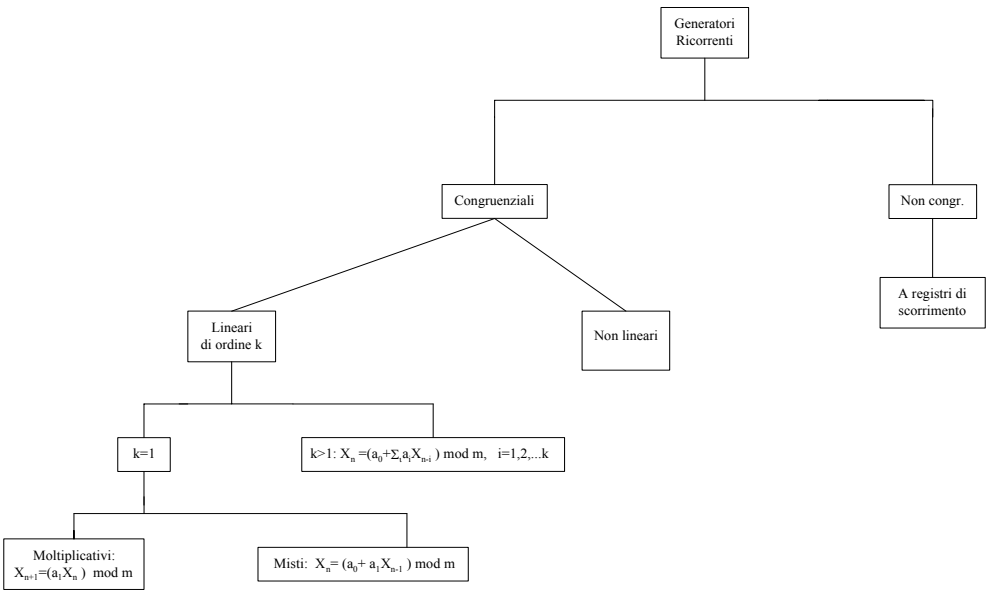
Lo schema che segue sintetizza le diverse tipologie di generatori classificati secondo alcuni criteri quali, ad esempio, la complessità e il numero di passi.

Nei paragrafi successivi verranno analizzati alcuni generatori ricorrenti sotto elencati.

Nonostante si sappia che i generatori più complessi riportati nel diagramma siano più efficienti del generatore moltiplicativo di ordine 1, perché soddisfano molte proprietà statistiche, tuttavia si preferisce utilizzare quest'ultimo per la semplicità e per la velocità [Fishman, 1996, pag. 588].

Nel software, comunque, sono state anche proposte alcune varianti che contribuiscono ad aumentare la casualità della distribuzione generata.

Per uno studio più completo si vedano Fishman [1996, 2001], Dagpunar [1988] e Niederreiter[1992].



Par. 1.2 - LA v.c. UNIFORME

Come vedremo in seguito, tutti gli algoritmi utilizzati per la generazione di v.c. continue e discrete trasformano adeguatamente la v.c. Uniforme $U(0,1)$ per ottenere la funzione di distribuzione desiderata.

Da ciò è possibile dedurre che un buon generatore di determinazioni della v.c. $U(0,1)$ è una condizione preliminare necessaria per garantire successivamente l'affidabilità dei risultati prodotti dagli algoritmi specifici che generano v.c. non Uniformi.

Richiamiamo brevemente alcune definizioni e proprietà della v.c. Uniforme.

Una v.c. Uniforme definita nell'intervallo generico (a,b) ha la seguente funzione di densità:

$$f_X(x) = \begin{cases} 0 & x \le a \\ \dfrac{1}{b-a} & a < x < b \\ 0 & x \ge b, \end{cases}$$

la corrispondente funzione di ripartizione è:

$$F_X(x) = \begin{cases} 0 & x \le a \\ \dfrac{x-a}{b-a} & a < x < b \\ 1 & x \ge b, \end{cases}$$

che è continua nell'intervallo (a,b).

Queste due funzioni così definite individuano in maniera univoca la distribuzione Uniforme.

I momenti di ordine *r* rispetto all'origine sono:

$$\mu_r(x) = \frac{1}{b-a}\left(\frac{b^{r+1} - a^{r+1}}{r+1}\right).$$

Pertanto i due primi momenti risultano essere:

$$\mu_1(x) = \frac{a+b}{2}$$
$$\mu_2(x) = \frac{b^3 - a^3}{(b-a)3} = \frac{b^2 + a^2 + ab}{3} \quad ,$$

da cui si ottiene il momento centrale di ordine *2* o varianza:

$$\sigma^2(x) = \frac{(b-a)^2}{12}.$$

La v.c. Uniforme standardizzata è:

$$Z = \frac{X - E(X)}{\sigma(X)} = \frac{\left(X - \frac{a+b}{2}\right)}{b-a} 2\sqrt{3}.$$

Nei problemi di simulazione si considera la v.c. Uniforme *U(0,1)* o Rettangolare con funzione di densità:

$$f_X(x) = \begin{cases} 1 & 0 < x < 1 \\ 0 & \text{altrove.} \end{cases}$$

Par. 1.3 - DEFINIZIONE E PROPRIETA' DI ALCUNI GENERATORI CONGRUENZIALI

Par.1.3.1- GENERATORI CONGRUENZIALI

Un modo efficace per ottenere sequenze di numeri pseudocasuali è dato dai generatori lineari congruenziali. Una famiglia di generatori di numeri pseudocasuali equidistribuiti tra (*0,m-1*] è quella dei generatori lineari congruenziali di ordine *k*:

$$X_n = \left(a_0 + a_1 X_{n-1} + a_2 X_{n-2} + \cdots + a_k X_{n-k}\right) m \text{ od } m \qquad 0 \le X_i \le m-1,\ 0 \le a_i \le m-1,$$

in cui X_i e a_i sono valori interi non negativi. La precedente relazione di congruenza, applicando l'operazione modulo che sarà descritto successivamente, può anche essere espressa in maniera equivalente:

$$X_n = \sum_{i=0}^{k} a_i X_{n-i} - k_n m, \quad \text{posto } X_{n-0} = 1,\ n = 1, 2, ...,$$

ove $k_n = \left\lfloor m^{-1} \sum_{i=0}^{k} a_i X_{n-i} \right\rfloor$ con periodo $m^k - 1$ [Fishman, 1996, pag. 646].

Il più noto, comunque, è il generatore misto di ordine 1:

$$X_n = \left(a_0 + a_1 X_{n-1}\right) \bmod m.$$

Questo generatore, che è anche il più diffuso, fu inizialmente proposto da Lehmer (1951) con $a_0 = 0$. ed è come generatore congruenziale moltiplicativo che è il più veloce di tutti i generatori congruenziali.

Successivamente, altri autori [Franklin; 1958; Thomson, 1958; Rotenberg, 1960; Greenberger, 1961a] hanno considerato anche il caso di $a_0 \neq 0$. Per una completa bibliografia relativa alla generazione di numeri pseudocasuali, si vedano Jansson [1966] e Sowey [1978].

Tutti i generatori congruenziali si basano sui principi della Teoria delle Congruenze di cui richiamiamo alcuni fondamenti per poter meglio comprendere la loro struttura [Ore, 1976; Dudley, 1978; Nagel, 1981; Maturo, 1989; de Resmini, 1993].

Di seguito sono riportati alcuni teoremi e proprietà sulle congruenze; per maggiori dettagli teorici si veda la bibliografia citata e in particolare Maturo.

Per valori interi positivi *a* e *b*, definiamo *mod* una operazione binaria:

$$a \bmod b \;\; = a - b\lfloor a/b \rfloor ,$$

dove per $\lfloor a/b \rfloor$ si intende l'intero inferiore più vicino al quoziente del rapporto tra *a* e *b;* l'intera espressione rappresenta il resto del quoziente tra *a* e *b*.

Quando *a* o *b* sono negativi, bisogna prestare attenzione al significato di *minimo intero.*

Ad esempio:

$5 \bmod 3 = 5 - 3\lfloor 5/3 \rfloor = 2$; $-5 \bmod 3 = -5 - 3\lfloor -5/3 \rfloor = 1$; $5 \bmod (-3) = 5 - (-3)\lfloor 5/(-3) \rfloor = -1$,

dove $\lfloor -5/3 \rfloor = -2$, perché $\frac{-5}{3} = -1.666$, e quindi il minimo intero è *–2*.

La definizione di modulo lascia non definita la divisione per *0.* Per convenzione si pone: $a \bmod 0 = a$ [Graham et. al., 1990, pag. 82].

La relazione di congruenza [Gauss, 1986] è così definita:
se *a*, *b*, *m* sono interi, *a* si dice congruo a *b* modulo *m* e si scrive:

$$a \equiv b \mod m , \tag{1}$$

se l'intero *a-b* è multiplo di *m*; oppure si dice congruo se:

$$a \bmod m = b \mod m , \tag{1a}$$

cioè gli interi *a* e *b*, rispetto al modulo *m*, hanno lo stesso resto.
Mentre se

$$a \equiv b \bmod m \quad \text{con } 0 \le b < m.$$

b si definisce *minimo residuo.*

Esempio*: 9 ≡ -16 mod 5* o anche *9-(-16)* è multiplo di *5 per la(1),* in modo analogo:

9 mod 5= -16 mod 5=4 per la (1a).

Sono inoltre valide le seguenti relazioni:

$$\begin{cases} (a \bmod m) \bmod m = a \bmod m \\ 0 \le a \bmod m < m \quad \text{se } m > 0 \\ 0 \ge a \bmod m \quad \text{se } m < 0. \end{cases} \tag{1b}$$

Definiamo $M.C.D.(m,a) = max\{k : \text{e } k \mid a\}$, dove $k|a$ indica che *k* divide *a*. Per convenzione *M.C.D.(m,0)=m.*

TEOREMA 1:

La congruenza rispetto a un determinato modulo *m* è una relazione di equivalenza, per cui valgono le seguenti proprietà:

1. Riflessiva $a \equiv a \bmod m$
2. Simmetrica $a \equiv b \bmod m \Rightarrow b \equiv a \bmod m$
3. Transitiva $a \equiv b \bmod m$ e $b \equiv c \bmod m \Rightarrow a \equiv c \bmod m.$