IFT 3245
Simulation et modèles

Fabian Bastin
DIRO
Université de Montréal

Automne 2016

Générateur récursif multiple (MRG)

Nous pouvons généraliser la récurrence du GCL par

$$x_n = (a_1 x_{n-1} + \cdots + a_k x_{n-k}) \mod m, \quad u_n = x_n/m.$$

En pratique, on prendra plutôt $u_n = (x_n + 1)/(m + 1)$, ou encore $u_n = x_n/(m + 1)$ si $x_n > 0$ et $u_n = m/(m + 1)$ sinon, mais la structure demeure essentiellement la même.

Si $k = 1$, nous retrouvons le générateur à congruence linéaire classique, avec $c = 0$.

L'état à l'étape n est $s_n = \boldsymbol{x}_n = (x_{n-k+1}, \ldots, x_n)^T$.
Espace d'états: $\mathcal{Z}_m^k$, de cardinalité m^k.
La période maximale est $\rho = m^k - 1$, pour m premier.

Polynôme caractéristique

On associe au MRG le polynôme caractéristique:

$$P(z) = z^k - a_1 z^{k-1} - \cdots - a_k = -\sum_{j=0}^{k} a_j z^{k-j},$$

où $a_0 = -1$.

Pour $k > 1$, pour avoir une période maximale, il est possible de montrer qu'il suffit d'avoir au moins deux coefficients non nuls, dont a_k. Ainsi, la récurrence la plus économique a la forme:

$$x_n = (a_r x_{n-r} + a_k x_{n-k}) \mod m,$$

avec $0 < r < k$.

$m = 2^e$

Une erreur fréquente, commise en particulier par les informaticiens peu au fait des statistiques, est de considérer $m = 2^e$.

Utiliser une puissance de 2 pour m permet en effet de facilement calculer le produit $ax \mod m$, et est parfois décrit comme efficace, ce qui est vrai du point de la rapidité d'exécution.

Les effets sur la période sont pourtant dommageables, vu que

- pour $k = 1$ et $e \geq 4$, on a $\rho \leq 2^{e-2}$;
- pour $k > 1$, on a $\rho \leq (2^k - 1)2^{e-1}$.

$m = 2^e$: exemple

Si $k = 7$ et $m = 2^{31} - 1$, la période maximale est $(2^{31} - 1)^7 - 1 \approx 2^{217}$. Mais pour $m = 2^{31}$ on a $\rho \leq (2^7 - 1)2^{31-1} < 2^{37}$, i.e. 2^{180} fois plus petit!

Pire, si nous nous intéressons au i^{th} bit le moins significatif, pour $k = 1$, la période de $x_n \mod 2^i$ ne peut pas dépasser $\max(1, 2^{i-2})$. Pour $k > 1$, la période de $x_n \mod 2^i$ ne peut pas dépasser $(2^k - 1)2^{i-1}$.

$m = 2^e$: exemple

Récurrence $x_n = 10205 x_{n-1} \mod 2^{15}$:

$$
\begin{aligned}
x_0 &= 12345 &= 011000000111001_2\\
x_1 &= 20533 &= 101000000110101_2\\
x_2 &= 20673 &= 101000011000001_2\\
x_3 &= 7581 &= 001110110011101_2\\
x_4 &= 31625 &= 111101110001001_2\\
x_5 &= 1093 &= 000010001000101_2\\
x_6 &= 12945 &= 011001010010001_2\\
x_7 &= 15917 &= 011111000101101_2.
\end{aligned}
$$

$m = 2^e$

De tels générateurs restent populaires, mais sont à proscrire dans des simulations dignes de ce nom. Ainsi, la fonction `ran48` reste présente dans les librairies C standards BSD.

m	a	c	Source
2^{24}	1140671485	12820163	early MS VisualBasic
2^{31}	65539	0	RANDU (IBM)
2^{31}	134775813	1	early Turbo Pascal
2^{31}	1103515245	12345	`rand()` in BSD ANSI C
2^{32}	69069	1	VAX/VMS systems
2^{32}	2147001325	715136305	BCLP language
2^{35}	5^{15}	7261067085	Knuth (1998)
2^{48}	68909602460261	0	Fishman (1990)
2^{48}	25214903917	11	Unix's `rand48()`
2^{48}	44485709377909	0	CRAY system
2^{59}	13^{13}	0	NAG Fortran/C library

Générateurs à sous-suites multiples

Afin de pouvoir adéquatement représenter les différentes variables aléatoires, il peut être intéressants de pouvoir instancier des générateurs de variables aléatoires à volonté, et faire évoluer ceux-ci en parallèle, plutôt que d'utiliser un seul générateur et transformer les tirs dans les distributions voulues à la volée.

Nous voudrions pouvoir utiliser plusieurs fois un même générateur au sein d'un programme, mais en débutant avec des semences différentes afin de produire des suites aléatoires différentes.

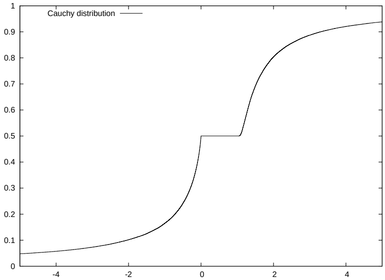
Générateurs à sous-suites multiples

Une première approche consiste à créer plusieurs générateurs, en spécifiant manuellement ces semences. Le danger majeur de cette approche est qu'il est difficile de prévoir la position des ces semences dans la séquence aléatoire, ce qui peut conduire à produire des séquences fortement corrélées. Le risque est d'autant plus élevé que la période du générateur est faible.

Exemple

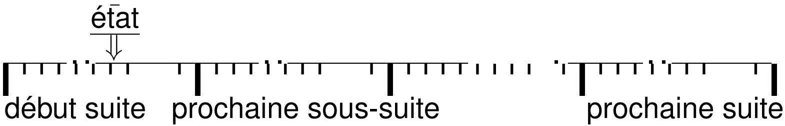
Soit X, Y, deux variables aléatoires normales $N(0,1)$ indépendantes. Il est possible de montrer que le rapport X/Y suit une distribution de Cauchy.

Générons ce rapport à l'aide du GCL Standard Minimal, avec 1 comme semence au numérateur, et 2 au dénominateur.



Générateurs à sous-suites multiples

Il est ainsi utile de pouvoir partitionner ces suites (ou “streams”) en sous-suites.



Sauts entre suites

Pour passer d'une suite à une autre, il est nécessaire de pouvoir calculer un point de la récurrence sans devoir générer tous les points intermédiaires. Or, nous pouvons écrire

$$\boldsymbol{x}_n = \boldsymbol{A}\boldsymbol{x}_{n-1} \mod m = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ a_k & a_{k-1} & \cdots & a_1 \end{pmatrix} \boldsymbol{x}_{n-1} \mod m.$$

Ainsi

$$\boldsymbol{x}_{n+\nu} = \boldsymbol{A}^{\nu}\boldsymbol{x}_n \mod m = (\boldsymbol{A}^{\nu} \mod m)\boldsymbol{x}_n \mod m.$$

Sauts entre suites

Nous pouvons précalculer $\boldsymbol{A}^{\nu} \mod m$ au moyen de la procédure suivante:

$$\boldsymbol{A}^{\nu} \mod m = \begin{cases} (\boldsymbol{A}^{\nu/2} \mod m)(\boldsymbol{A}^{\nu/2} \mod m) \mod m & \text{si } \nu \text{ est pair;} \\ \boldsymbol{A}(\boldsymbol{A}^{\nu-1} \mod m) \mod m & \text{si } \nu \text{ est impair.} \end{cases}$$