IFT 3245
Simulation et modèles

Fabian Bastin
DIRO
Université de Montréal

Automne 2016

Lois non uniformes

Supposons que nous disposons d'un bon générateur fournissant des variables aléatoires i.i.d. $U(0,1)$.

En pratique, nous souhaitons cependant générer des variables aléatoires plutôt selon des lois diverses (normale, Weibull, Poisson, binomiale, etc.), voire d'autres "objets" aléatoires: processus stochastiques, points sur une sphère, matrices aléatoires, arbres aléatoires, etc.

Pour ce faire, il convient donc de pouvoir transformer une uniforme $U(0,1)$ de manière adéquate.

Propriétés recherchées

- méthode correcte (ou très bonne approximation);
- simple: facile à comprendre et à implanter;
- rapide: après un temps d'initialisation ("setup"), si requis, le temps marginal par appel devrait être aussi faible que possible.
- mémoire utilisée;
- robustesse: l'algorithme doit être précis et efficace pour toutes les valeurs des paramètres qui peuvent nous intéresser.

Propriétés recherchées

Nous privilégierons aussi les approches compatible avec les méthodes de réduction de variance.

Par exemple, nous préfèrerons habituellement l'inversion parce que cela facilite la synchronisation lorsque l'on compare deux système ou lorsqu'on veut utiliser des variables de contrôle, ou des valeurs antithétiques, etc.

Nous serons prêts à sacrifier un peu la vitesse pour préserver l'inversion.

Les méthodes de réduction de la variance nous feront au final souvent gagner beaucoup plus de temps de calcul que ce que l'on aura sacrifié pour préserver l'inversion.

Inversion: cas continu

La technique majeure pour générer une v.a. X est l'inversion.

Considérons la fonction de répartition F de X. Soit $U \sim U(0,1)$ et

$$X = F^{-1}(U) = \min\{x : F(x) \geq U\}.$$

Alors, dans le cas continu,

$$P[X \leq x] = P[F^{-1}(U) \leq x] = P[U \leq F(x)] = F(x),$$

i.e., X suit la fonction de répartition voulue.

Inversion: cas discret

Dans le cas discret, il s'agit de prouver que $P[X = x_i] = p(x_i)$, pour tout i, et on supposera $x_1 < x_2 < \ldots < x_n$.

Pour $i = 1$, nous avons $X = x_1$ si et seulement si $U \leq F(x_1) = p(x_1)$, comme voulu.

Pour $i \geq 2$, nous avons $X = x_i$ si et seulement si $F_{i-1} < U \leq F(x_i)$.

Comme $0 \leq F(x_{i-1}) < F(x_i) \leq 1$, nous obtenons

$$P[X = x_i] = P[F(x_{i-1}) < U \leq F(x_i)] = F(x_i) - F(x_{i-1}) = p(x_i).$$

Il est facile de prouver que le principe fonctionne toujours pour des distributions mixtes, c'est-à-dire ayant des composantes discrètes et continues.

Inversion: cas discret

La technique d'inversion a pour avantage d'être monotone: il y a un seul U pour chaque X.

Mais pour certaines lois, F peut est très difficile à inverser. Toutefois, nous pouvons souvent quand même approximer F^{-1} numériquement.

De manière générale, on peut utiliser des techniques d'interpolation (par exemple une interpolation de l'Hermite ou, dans le cadre de l'estimation de modèles, à l'aide de B-splines).

Exemple: Loi triangulaire

Pour générer une triangulaire définie sur l'intervalle $[a, b]$ et de mode c, par inversion, on commencera par tirer u suivant une uniforme $U(0, 1)$.

Il s'agit tout d'abord de déterminer si nous sommes à gauche ou à droite du mode. Partant de la fonction de densité

$$f(x) = \begin{cases} \frac{2}{b-a}\frac{x-a}{c-a} & \text{if } a \leq x \leq c, \\ \frac{2}{b-a}\frac{b-x}{b-c} & \text{if } c \leq x \leq b, \\ 0 & \text{otherwise.} \end{cases}$$

Nous avons que la probabilité d'obtenir un nombre inférieur à c est

$$\frac{2}{b-a}\int_a^c \frac{x-a}{c-a} = \left.\frac{x^2 - 2ax}{(b-a)(c-a)}\right|_a^c = \frac{c-a}{b-a}.$$

Exemple: Loi triangulaire

Comme la fonction de répartition est

$$\begin{cases} \frac{(x-a)^2}{(b-a)(c-a)} & \text{if } a \leq x \leq c, \\ 1 - \frac{(b-x)^2}{(b-a)(b-c)} & \text{if } c \leq x \leq b, \\ 0 & \text{otherwise.} \end{cases}$$

il suffit d'inverser cette fonction et par conséquent, de retourner

$$\begin{cases} a + \sqrt{(b-a)(c-a)u} & \text{si } u \leq \frac{c-a}{b-a}, \\ b - \sqrt{(b-a)(b-c)(1-u)} & \text{sinon.} \end{cases}$$

Exemple: Loi normale

Si $Z \sim N(0,1)$, alors $X = \sigma Z + \mu : N(\mu, \sigma^2)$. Il suffit donc de savoir générer une $N(0,1)$, de densité: $f(x) = (2\pi)^{-1/2} e^{-x^2/2}$.

Mais nous n'avons pas de formule pour $F(x)$ ni pour $F^{-1}(x)$.

Par contre, on sait que quand x est grand, $F(x)$ ressemble à

$$\tilde{F}(x) = 1 - \frac{1}{x\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

dont l'inverse[1] est $\tilde{F}^{-1}(u) = \sqrt{-2 \ln \sqrt{2\pi}(1-u)}$.

[1] Nous négligeons le terme $-\ln x$ comme x est grand, de sorte que l'expression est dominée par $-x^2/2$.

Exemple: Loi normale

L'idée de base est, pour $x > 0$, d'approximer $x = F^{-1}(u)$ par $y = \tilde{F}^{-1}(u)$, plus un quotient de 2 polynômes en y (i.e., une fonction rationnelle) qui approxime la différence.

Le cas $x < 0$ en découle directement en raison de la symétrie de $F^{-1}(\cdot)$ autour de 0.5.

Cela donne, si $U > 1/2$,

$$Y = \sqrt{-2ln[(1-U)\sqrt{2\pi}]},$$
$$X = Y + \frac{p_0 + p_1 Y + p_2 Y^2 + p_3 Y^3 + p_4 Y^4}{q_0 + q_1 Y + q_2 Y^2 + q_3 Y^3 + q_4 Y^4}$$

Les p_i et q_i sont choisis de manière à ce que l'approximation soit excellente pour tout $U > 1/2$. Si $U < 1/2$, on utilise la symétrie: on calcule X pour $1 - U$ au lieu de U, puis on retourne $-X$.

Wichura (1988) a ainsi proposé un algorithme basé sur ce principe qui permet de calculer la fonction de répartition inverse jusqu'aux environs de 16 chiffres significatif, soit de l'ordre de la précision machine, pour que autant $\min\{u, 1-u\}$ soit plus grand que 10^{-316}.

Pour des distributions telles que la χ^2, la gamma, la beta, etc., les choses se compliquent car la forme de F^{-1} depend des paramètres.

Implantation de l'inversion pour les lois discrètes

Rappelons que

$$p(x_i) = P[X = x_i];$$
$$F(x) = \sum_{x_i \leq x} p(x_i).$$

La technique d'inversion revient dès lors à générer U, trouver $I = \min\{i | F(x_i) \geq U\}$ et retourner x_I.

Implantation de l'inversion pour les lois discrètes

Initialisation: mettre les x_i et les $F(x_i)$ dans des tableaux, pour $i = 1, \ldots, n$.

1. Recherche linéaire (temps en $O(n)$):
   $U \leftarrow U(0,1)$; $\quad i \leftarrow 1$; tant que $F(x_i) < U$ faire $i \leftarrow i + 1$;
   retourner x_i.
2. Recherche binaire (temps en $O(\log(n))$):
   $U \leftarrow U(0,1)$; $\quad L \leftarrow 0$; $\quad R \leftarrow n$;
   tant que $L < R - 1$
   $\quad m \leftarrow \lfloor (L+R)/2 \rfloor$;
   $\quad$ si $F(x_m) < U$ alors $L \leftarrow m$ sinon $R \leftarrow m$;
   $\quad$ (* Invariant: l'indice I est dans $\{L+1, \ldots, R\}$. *)
   retourner x_R.

Autres approches: composition

Supposons que F est une combinaison convexe de plusieurs fonctions de répartition:

$$F(x) = \sum_{j=0}^{\infty} p_j F_j(x),$$

avec $\sum_{j=0}^{\infty} p_j = 1$, et qu'il est plus facile d'inverser F_j, $j = 0, \ldots, \infty$ que F. Un algorithme simple pour tirer de F dans pareil cas est décrit ci-dessous.

Génération par composition

Générer $J = j$ avec probabilité p_j, puis générer X selon F_J.

La méthode requiert dès lors deux uniformes pour chaque variable, et exploite la décomposition

$$P[X \leq x] = \sum_{j=0}^{\infty} P[X \leq x | J = j] P[J = j] = \sum_{j=0}^{\infty} F_j(x) p_j,$$

qui montre qu'il suffit de générer X en conditionnant sur la valeur de J.

Convolution

Supposons que la variable aléatoire X qui nous intéresse puisse s'écrire comme suit:

$$X = Y_1 + Y_2 + \ldots + Y_n,$$

où les Y_i sont indépendantes, de lois spécifiées.

Il suffit alors de générer les Y_i et de sommer.

Exemples: Erlang (somme d'exponentielles de même moyenne), binomiale.