Probabilités pour l'Informatique

Brice Olivier[1]

[1]Université Grenoble Alpes, Laboratoire Jean Kuntzmann et Inria
brice.olivier@inria.fr

Remerciements à J.-B. Durand, Y. Pigeonnat, H. Guiol

Ensimag 2A

Planning

- 18h de cours
- 12 séances de 1h30
- 2 séances par semaine
- Du 22 Sept. au 4 Nov.
- Evaluation : DS + TPs notés ou projet (à décider)

Outils

Ressources sur chamilo :
`http://chamilo2.grenet.fr/inp/courses/ENSIMAG4MM1PPI`

- Pré-requis et rappels
- Support de cours
- Exercices complémentaires
- Examens des années précédentes
- Probabilités et Python

Programmation sur python, ipython Notebook, Spyder, ...

Utilité du cours dans la formation / vie professionnelle

- Traitement du signal, théorie de l'information, compression
- Programmation, analyse et simulation de systèmes en présence d'aléa
- Evaluation de performances
- Compétence Maths/Info : valeur ajoutée

- Exemple : En Big Data, les Resources and Jobs Management Systems (RJMS) ont besoin d'évoluer en affectant constamment des ressources et des jobs aux différentes entités. Cependant, l'expérimentation sur ce genre de système s'avère extrêmement couteuse en termes d'énergie et de période d'arrêt. Beaucoup d'algorithmes d'ordonnancement théoriques émergent mais ne voient pas le jour car trop peu réalistes. L'idée serait de pouvoir **simuler de tels systèmes** en amont pour tester leur efficacité pratique. (Cf. Batsim)

Outline

Intuitions I

- Qu'est ce qu'un générateur de nombres aléatoires ? Comment le mettre en oeuvre sur machine ?
- Quelles propriétés intéressantes peut-il avoir ? Comment les vérifier ?

- Travail individuel de reflexion
- Débat intra-groupe (3 personnes)
- Débat inter-groupe

Intuitions II

Les nombres générés doivent être :

- uniformément distribués
- mutuellement indépendants
- reproductibles : toute simulation doit pouvoir être reproduite par n'importe quel utilisateur en utilisant le même jeu de données

Générateur congruentiel linéaire

Définition du générateur congruentiel linéaire

Modèle

$$x_{n+1} = (a * x_n + c) \mod m, \qquad n \geq 0$$

le module $m \in \mathbb{N}^*$,

le multiplieur $a \in \mathbb{N} \quad \cap \quad 0 < a < m$,

l'incrément $c \in \mathbb{N} \quad \cap \quad 0 \leq c < m$,

la graine $x_0 \in \mathbb{N} \quad \cap \quad 0 \leq x_0 < m$

Exemple

Calculer $x_0, ..., x_{10}$ avec $m = 10, a = c = x_0 = 7$

Générateur congruentiel linéaire

Définition du générateur congruentiel linéaire

Modèle

$$x_{n+1} = (a * x_n + c) \mod m, \qquad n \geq 0$$

le module $m \in \mathbb{N}^*$,

le multiplieur $a \in \mathbb{N} \quad \cap \quad 0 < a < m$,

l'incrément $c \in \mathbb{N} \quad \cap \quad 0 \leq c < m$,

la graine $x_0 \in \mathbb{N} \quad \cap \quad 0 \leq x_0 < m$

Exemple

Calculer $x_0, ..., x_{10}$ avec $m = 10, a = c = x_0 = 7$

Séquence congruentielle linéaire obtenue : $\{7, 6, 9, 0, 7, 6, 9, 0, 7, 6\}$

Générateur congruentiel linéaire

Quelques remarques

- La séquence congruentielle linéaire se répète au bout d'un certain nombre d'itérations
- La longueur de la séquence repétée s'appelle **la période** de longueur $d \leq m$
- Dans l'exemple précédent, $\{7, 6, 9, 0, 7, 6, 9, 0, 7, 6\}$, la période est de longueur 4
- Evidemment, Une bonne séquence sera de période d relativement longue
- Un générateur est dit de **pleine période** si quelque soit x_0, il produit des séquences de période de longueur m

Comment choisir a, c, m tels que $\forall x_0, d = m$?

Générateur congruentiel linéaire

Exercice - choix des paramètres a, c, m

Par groupe de 5

- Pour $m = 11, c = 0, a \in \{1, ..., 6\}$, trouver les a qui mènent à un générateur de pleine période
- Le générateur $m = 10, c = 3, a = 1$ est-il de pleine période ?
- Le générateur $m = 10, c = 3, a = 21$ est-il de pleine période ?
- Quelles propriétés pouvez-vous en déduire quant au choix des paramètres a, c, m ?

Générateur congruentiel linéaire

Choix des paramètres a, c, m

Pour $c \neq 0$, une séquence congruentielle linéaire définie par les paramètres m, a, c, x_0 est de pleine période quelque soit x_0 si et seulement si :

- c et m sont premiers entre eux $\iff PGCD(c, m) = 1$
- Pour chaque nombre premier p_i divisant m, $a - 1$ est multiple de p_i
- si m est un multiple de 4, alors $a - 1$ est un multiple de 4

Pour $c = 0$, une séquence congruentielle linéaire définie par les paramètres m, a, x_0 est de pleine période $(m - 1)$ quelque soit x_0 si et seulement si :

- m est premier
- $a^{m-1} - 1$ est un multiple de m
- $\forall i = 1, ..., m - 2, a^i - 1$ n'est pas divisible par m
- Cas particulier lorsque $x_0 = 0$

Générateur congruentiel linéaire

Exercices - choix du paramètre m

Par groupe de 5

- Pour $m = 16$, déterminer quelques couples a, c tel(s) que le(s) générateur(s) produit(s) est/sont de pleine période
- Pour $m = 32$, déterminer le(s) a, c tel(s) que le(s) générateur(s) produit(s) est/sont de pleine période
- Astuce : Si m est une puissance de 2, il suffit de choisir c impair et $a = 4n + 1, \forall n \in \mathbb{N}^*$
- Que remarquez-vous ?

Générateur congruentiel linéaire

Choix de m

Critères pour le choix de m :

1. **Augmenter la période** p
2. **Rapidité de génération** des nombres aléatoires

Solutions :

1. Pour augmenter p, il suffit d'augmenter m
2. Une astuce est de choisir $m = 2^e, e \in \mathbb{N}^*$.
   En effet, calculer $a * x_n \mod {2^e}_{(10)}$ équivaut à conserver les e bits de poids faibles de $a * x_{n(2)}$, puis à reconvertir le résultat en décimal

Générateur congruentiel linéaire

Problème d'implémentation

Problème : le calcul de $a * x_n$ peut résulter en un overflow mémoire

Méthode 1 :
Mettre a sous forme $a = 2^\alpha a_1 + a_2$, avec $a_1, a_2 < 2^\alpha$
On a alors,

$$ax \mod m = (a_1 * (2^\alpha x \mod m) + a_2 x \mod m) \mod m$$

Méthode 2 :
Si $a < \sqrt{m}$, on pose $q = \lfloor \frac{m}{a} \rfloor$ (partie entière) et $r = m \mod a$, on peut donc écrire $m = aq + r$ On a alors,

$$ax \mod m = a(x \mod q) - \lfloor \frac{x}{q} \rfloor r + (\lfloor \frac{x}{q} \rfloor - \lfloor \frac{ax}{m} \rfloor)m$$

Astuce : le terme de droite vaut 0 si $a(x \mod q) - \lfloor \frac{x}{q} \rfloor r \geq 0$

Générateur congruentiel linéaire

Qualités et défauts (1/2)

Qualités

- Facilité d'implémentation
- Génération très rapide des nombres
- Peu gourmand en termes de mémoire

Défauts

- Mauvaise qualité de l'aléa (cryptographie)
- Structure en treillis des données successives

Générateur congruentiel linéaire

Qualités et défauts (2/2)

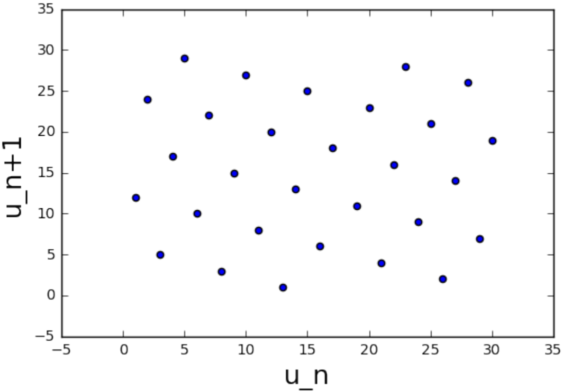

Figure – Structure en treillis des données successives ($a = 12, c = 0, m = 31, x_0 = 30$)

Générateur congruentiel linéaire

Générateurs combinés

L'idée est de combiner plusieurs des générateurs congruentiels linéaires

On prend $J \geq 2$ générateurs linéaire de la forme :

$$x_{j,i+1} = a_j x_{j,i} \mod m_j, j = 1, ..., J$$

Puis on calcule le terme suivant de la suite :

$$x_{i+1} = \sum_{j=1}^{J} (-1)^{j-1} x_{j,i+1} \mod (m^* - 1)$$

où $m^* = max(m_j)$

Générateur congruentiel linéaire

Générateurs à récursivité multiple

L'idée est d'utiliser les $k-1$ valeurs précédentes pour générer la valeur courante

On a la suite définie par :

$$x_{i+1} = (a_1 x_i + a_2 x_{i-1} + ... + a_k x_{i-k+1}) \mod m$$

avec une seed du type : $x_{k-1}, ..., x_0$

Etant donné que chaque x_i peut avoir m valeurs distinctes, le vecteur $(x_{i-1}, ..., x_{i-k})$ peut avoir m^k valeurs distinctes. La période maximale est alors m^{k-1}

Pour les conditions sur m et a_j, Cf. (Knuth 1998)

Introduction

Maintenant que nous disposons d'un générateur pseudo-aléatoire, supposons le GCL, comment simuler des lois de probabilités à partir de celui-ci ?

Pour commencer, on se ramène à la loi la plus simple, la loi uniforme $\mathcal{U}(0,1)$

$$u_n = x_n/m$$

u_n sont des réalisations des variables aléatoires U_n. Etant donné que $\forall i \neq j, U_i \perp U_j$, alors $\forall n, U = U_n$, on a donc $U \sim \mathcal{U}(0,1)$

Objet du chapitre : Comment utiliser U pour simuler une variable aléatoire X de loi donnée P_x (densité f_X, fonction de répartition F_X)

Quelques rappels

- F_X la fonction de répartition de la variable aléatoire X, t.q. $F_X(x) = P(X \leq x)$
- Pour tout intervalle, on a donc $P(x \leq X < x + h) = F_X(x + h) - F_X(x)$
- Si la fonction de répartition est continue, alors on a : $F_x(x) = \int_{-\infty}^{X} f_X(u)du$, où f_X est la fonction de densité
- $f_X(x) = lim_{\delta x \to 0} P(x \leq X < x + \delta x)\delta x$
- Axiomes de probabilités : $\int_{-\infty}^{+\infty} f_X(x)dx = 1$ et $0 \leq P(x) \leq 1$
- (+ Examples pdf, cdf : uniform, exponential, gaussian)

Exemple introductif

Par petits groupes :

- Comment simuler un lancé de dé à 6 faces à partir d'une loi uniforme $\mathcal{U}(0,1)$?

Exemple introductif

Par petits groupes :

- Comment simuler un lancé de dé à 6 faces à partir d'une loi uniforme $\mathcal{U}(0,1)$?

$$X = \begin{cases} X = 1 \text{ si } 0 \leq U < \frac{1}{6} \\ X = 2 \text{ si } \frac{1}{6} \leq U < \frac{2}{6} \\ X = k \text{ si } \frac{k-1}{6} \leq U < \frac{k}{6} \end{cases}$$

$X \in \{1, ..., 6\}$

$P(X = k) = P(U \in [\frac{k-1}{6}; \frac{k}{6}]) = \frac{k}{6} - \frac{k-1}{6} = \frac{1}{6}$

(+ Fonction de répartion)

Simulation par inversion

Propriété : Si $U \sim \mathcal{U}(0,1)$, alors $F^{-1}(U) \sim P_x$
($X = F_X^{-1}(U)$ a pour fonction de répartition F_x)

(+ Exemple loi normale)

Exercices : A partir d'une loi uniforme $\mathcal{U}(0,1)$, simuler

- Loi de Bernoulli de paramètre $p = 0.6$
- Loi de Poisson de paramètre $\lambda = 6$
- Loi Exponentielle de paramètre $\lambda = 0.5$
- Loi Binomiale de paramètre $n = 10, p = 0.6$, puis $n = 1000, p = 0.001$

Références I

- Cours Probabilités pour l'informatique de Jean-Baptiste Durand (2015-2016)
- Cours Probabilités pour l'informatique de Hervé Guiol (2012)
- Cours de Probabilités Appliquées, Olivier François
- Testing Random-Number Generators, Raj Jain, Washington University
- Cours de simulation de lois, Christine HEINEMANN, HEC Paris
- Monte Carlo Methods in Financial Engineering, Paul Glasserman