Analyse d'algorithmes, génération aléatoire de vecteurs stochastiques ... et entropie

Auteur: Julien DAVID

De quoi on va parler?

Problème Génération aléatoire de vecteurs stochastiques...

On s'intéresse aux vecteurs stochastiques dont l'image par une fonction donnée prend une valeur donnée.

Génération aléatoire d'objets combinatoires

Définition — Classe combinatoire

Une classe combinatoire est un ensemble d'objets $\mathcal{O}$ muni d'une fonction de taille tel que l'ensemble $\mathcal{O}_n \subseteq \mathcal{O}$ des objets de taille n est **fini**.

Génération aléatoire d'objets combinatoires

Définition Classe combinatoire

Une classe combinatoire est un ensemble d'objets $\mathcal{O}$ muni d'une fonction de taille tel que l'ensemble $\mathcal{O}_n \subseteq \mathcal{O}$ des objets de taille n est **fini**.

Définition Générateur aléatoire

Soit π_n distribution de probabilité sur $\mathcal{O}_n$, un générateur aléatoire est un algorithme qui produit un objet de $\mathcal{O}_n$ selon π_n.

Exemple - Lien avec l'exposé précédent

Exemple Distribution sur du texte

- On considère des sources discrètes sans mémoire.
- Soit un alphabet à k lettres Σ et une distribution $\pi = (\pi_1, \ldots, \pi_k)$.
- Les mots produits de longueurs n suivent une distribution $\pi_{\otimes n}$ induite par π.

Algorithme 1 : Générateur aléatoire

Data : Une distribution π, un longueur n
Result : Texte w de longueur n
pour $i \in \{1, \cdots, n\}$ **faire**
  $w_i =$ une lettre tirée aléatoirement selon π;
fin
return w

Pourquoi donc?

Définition Complexité algorithmique

- Soit un algorithme prenant en entrée un objet combinatoire de taille n.
- Soit la fonction $Cout : \mathcal{O}_n \mapsto \mathbb{N}$ qui prend en entrée un objet combinatoire et renvoie le nombre d'instructions effectuées par l'algorithme sur cette entrée.
- La complexité **dans le pire des cas** d'un algorithme est l'estimation de

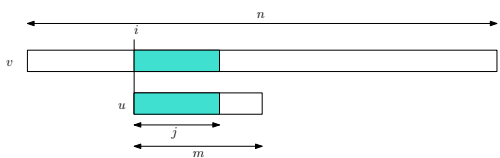
$$\lim_{n \to \infty} \max_{o \in \mathcal{O}_n} \{Cout(o)\}$$

Exemple: Recherche d'un motif dans un texte

Algorithme 1 : Algorithme Naïf

Data : Texte v de longueur n, mot u de longueur m
Result : Nombre d'occurrences de u dans v

```
occ ← 0;
pour i ∈ {1, ··· , n − m + 1} faire
    j ← 1;
    tant que v_{i+j−1} = u_j and j ≤ m faire
        j ← j + 1;
    fin
    si j = m + 1 alors
        occ ← occ + 1;
    fin
fin
return occ
```



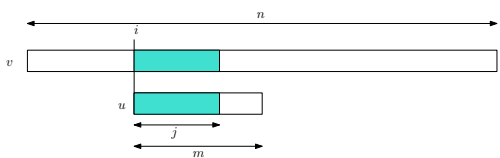
Complexité pire des cas: $\Theta(nm)$

Exemple: Recherche d'un motif dans un texte

Algorithme 1 : Algorithme Naïf

Data : Texte v de longueur n, mot u de longueur m
Result : Nombre d'occurrences de u dans v

```
occ ← 0;
pour i ∈ {1, ··· , n − m + 1} faire
    j ← 1;
    tant que v_{i+j−1} = u_j and j ≤ m faire
        j ← j + 1;
    fin
    si j = m + 1 alors
        occ ← occ + 1;
    fin
fin
return occ
```



Complexité pire des cas: $\Theta(nm)$
Complexité meilleur des cas: $\Theta(n)$

Pourquoi donc?

Définition Complexité algorithmique **en moyenne**

- Soit un algorithme prenant en entrée un objet combinatoire de taille n.
- Soit $CoutMoyen$ la fonction qui prend en entrée une distribution π_n sur les entrées d'un algorithme et renvoie le coût moyen de cet algorithme selon cette distribution.
- La complexité **moyenne** d'un algorithme est l'estimation de

$$\lim_{n \to \infty} CoutMoyen(\pi_n)$$

Génération aléatoire et analyse d'algorithme

- Si l'on possède un générateur aléatoire de $\mathcal{O}_n$ selon une distribution π_n ...

Génération aléatoire et analyse d'algorithme

- Si l'on possède un générateur aléatoire de $\mathcal{O}_n$ selon une distribution π_n ...
- il est possible d'étudier expérimentalement $CoutMoyen(\pi_n)$

Génération aléatoire et analyse d'algorithme

- Si l'on possède un générateur aléatoire de $\mathcal{O}_n$ selon une distribution π_n ...
- il est possible d'étudier expérimentalement $CoutMoyen(\pi_n)$
- C'est donc un outil très pratique pour les chercheuses et les chercheurs.

Exemple: Recherche d'un motif dans un texte

Algorithme 1 : Algorithme Naïf

Data : Texte v de longueur n, mot u de longueur m
Result : Nombre d'occurrences de u dans v

```
occ ← 0;
pour i ∈ {1, ··· , n − m + 1} faire
    j ← 1;
    tant que v_{i+j−1} = u_j and j ≤ m faire
        j ← j + 1;
    fin
    si j = m + 1 alors
        occ ← occ + 1;
    fin
fin
return occ
```

Pour la distribution **uniforme** sur les mots de taille n et m, la complexité moyenne est $\Theta(n)$

Théorie VS pratique

- En pratique, l'algorithme se comporte comme le prédit la complexité moyenne.
- Il est pourtant assez évident que les textes en langages naturels diffèrent de ceux produits par un générateur aléatoire uniforme.
- Les "distributions réelles" et l'uniforme ont, il semblerait, une propriété commune, déterminante pour le comportement de l'algorithme.

Point de vue nouveau

Problème Analyse d'algorithme sur un ensemble de distribution

- Soit $CoutMoyen$ la fonction qui prend en entrée une distribution sur les entrées d'un algorithme et renvoie le coût moyen de cet algorithme selon cette distribution.
- Soit D un ensemble de distributions (ayant de préférence une propriété commune).
- On souhaite étudier l'intégrale suivante:

$$\int_D CoutMoyen(\pi)\ d\pi$$

Point de vue nouveau

Problème Analyse d'algorithme sur un ensemble de distribution

- Soit $CoutMoyen$ la fonction qui prend en entrée une distribution sur les entrées d'un algorithme et renvoie le coût moyen de cet algorithme selon cette distribution.
- Soit D un ensemble de distributions (ayant de préférence une propriété commune).
- On souhaite étudier l'intégrale suivante:

$$\int_D CoutMoyen(\pi)\ d\pi$$

Pour faciliter l'étude expérimentale, on veut donc des générateurs aléatoires d'éléments de D

Propriétés sur les distributions

- Clément, Nguyen Thi et Vallée ont étudiés 3 algorithmes de tris classiques.
- iels démontrent que leur complexité moyenne dépend des propriétés suivantes:

Algorithmes	**Propriétés sur les sources**	**Terme Dominant**
Tri Rapide	Entropie de Shannon	$\frac{1}{h(\pi)} n \log^2 n$
Tri Insertion	Entropie de Shannon Entropie de collision	$\frac{c(\pi)}{4} n^2$
Tri à Bulles	Entropie de Shannon Entropie de collision	$\frac{1}{4h(\pi)} n^2 \log n$

Entropie des distributions

Exemple — Propriété sur les distributions

On va donc considérer des ensembles D de distributions ayant

- soit la même entropie de Shannon,
- soit la même entropie de collision,

et fabriquer un générateur aléatoire d'éléments de D.

Entropie des distributions

Exemple Propriété sur les distributions

On va donc considérer des ensembles D de distributions ayant

- soit la même entropie de Shannon,
- soit la même entropie de collision,

et fabriquer un générateur aléatoire d'éléments de D.

Afin d'éviter toute confusion, on appelle les éléments de D des **vecteurs sources**.

Généralisation des vecteurs stochastiques

Définition Vecteur source

Soit une constante $0 < s \leq 1$, un vecteur source de dimension k est une séquence de valeurs $(x_1, \ldots, x_k)$ telle que

$$\sum_{i=1}^{k} x_i = s$$

Ensemble de vecteurs sources discrets

Remarque — Vecteurs sources discrets

On considère qu'une valeur x_i est un multiple d'une valeur minimum ε.

Ensemble de vecteurs sources discrets

Remarque — Vecteurs sources discrets

On considère qu'une valeur x_i est un multiple d'une valeur minimum ε.

Définition — Ensemble de vecteurs sources

Soient $0 < s \leq 1$, une dimension k et une valeur minimum ε, l'ensemble fini des vecteurs sources que l'on considère est

$$\mathbb{S}_{k,\varepsilon}(s) = \{(x_1, \ldots, x_k) \in (\varepsilon \mathbb{N}^{\geq 1})^{\,k} \mid \sum_{i=1}^{k} x_i = s\}$$

Vecteurs sources dont une propriété est fixée.

Objectif Entropie

- On veut se resteindre à un sous-ensemble de $\mathbb{S}_{k,\varepsilon}(s)$ dont l'entropie est égale à une valeur t
- On veut ensuite pouvoir tirer aléatoirement et uniformément un élement de ce sous-ensemble.

Vecteurs sources dont une propriété est fixée.

Objectif — Entropie

- On veut se resteindre à un sous-ensemble de $\mathbb{S}_{k,\varepsilon}(s)$ dont l'entropie est égale à une valeur t
- On veut ensuite pouvoir tirer aléatoirement et uniformément un élement de ce sous-ensemble.

Objectif — Ce serait dommage d'en rester là

On va décrire une famille de fonctions pour lesquelles la méthode fonctionne.

En admettant que je mets des choses sous le tapis...

Soit $T_k : \mathbb{S}_{k,\varepsilon}(s) \mapsto \mathbb{R}$ une fonction **symétrique, unimodale et concave**

Objectif Génération aléatoire

Pour un k, ε, t fixé, on souhaite engendrer aléatoirement et uniformément un élément de

$$\mathbb{S}_{k,\varepsilon,t}(s) = \{(x_1, \ldots, x_k) \in \mathbb{S}_{k,\varepsilon}(s) \mid T(x_1, \ldots, x_k) = t\}$$

Exemples: fonctions symétriques, unimodales et concave

Définition — Entropie de Shannon

$$T_k(\pi) = -\sum_{i=1}^{k} \pi_i \cdot \log_2(\pi_i)$$

Définition — Produit

$$T_k(\pi) = \prod_{i=1}^{k} \pi_i$$

Définition — Entropie de collision

$$T_k(\pi) = -\log_2(\sum_{i=1}^{k} \pi_i^2)$$

Définition — Moyenne géométrique

$$T_k(\pi) = \left(\prod_{i=1}^{k} \pi_i\right)^{\frac{1}{k}}$$

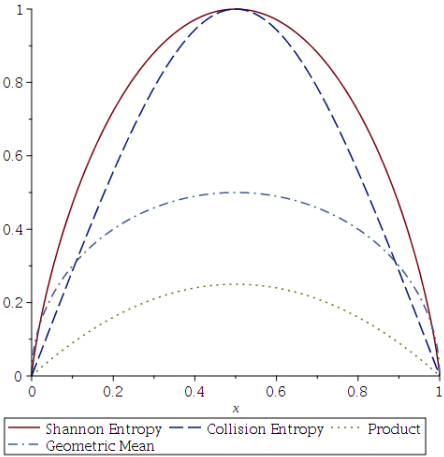
Exemples: fonctions symétriques, unimodales et convexe

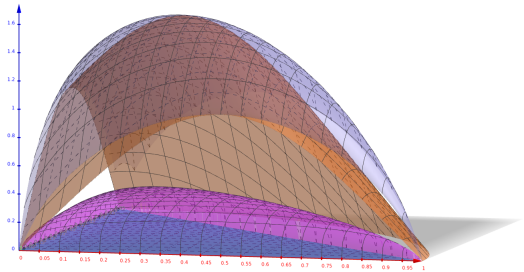
Exemple Bonus!!!

La méthode que je vais présenter peut-être adaptée pour des fonctions convexes comme:

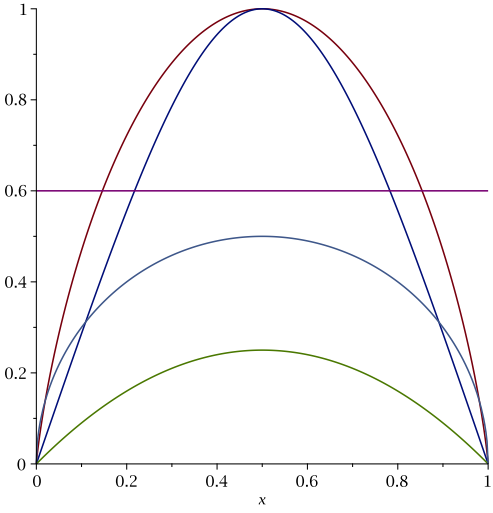
- la distance totale de variation à l'uniforme.
- la divergence de Kullback-Leibler à l'uniforme.

Fonctions symétriques, unimodales, concaves

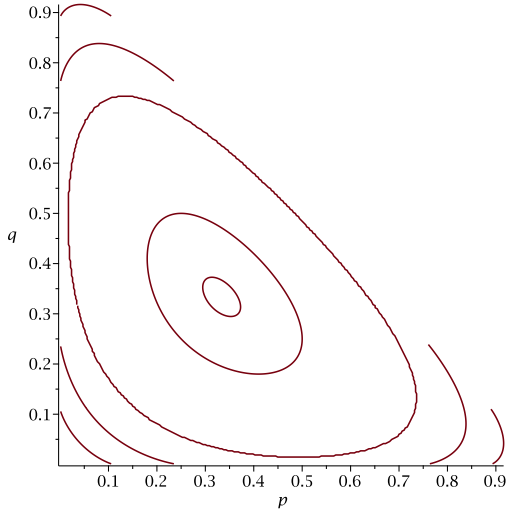




Ensembles $\mathbb{S}_{k,\varepsilon,t}(1)$



Dimension 2: 0, 1 ou 2 solutions.



Dimension 3: pas nécessairement connexe.

Minimum et maximum

Remarque Minimum et maximum

Soient k variables de somme $s \leq 1$, une fonction symétrique unimodale concave est maximale pour

$$T_k(\frac{s}{k}, \ldots, \frac{s}{k})$$

Si la valeur minimum des variables est ε, la fonction est minimale pour

$$T_k(s - (k-1)\varepsilon, \varepsilon, \ldots, \varepsilon)$$

En dimension 2

Remarque — Le cas particulier de la dimension 2

L'ensemble

$$\{(x_1, x_2) \in \mathbb{S}_{2,\varepsilon}(s) \mid T_2(x_1, x_2) = t\}$$

ne contient

- aucun élément si $t > T_2(\frac{s}{2}, \frac{s}{2})$ ou $t < T_2(s - \varepsilon, \varepsilon)$
- un seul élement si $t = T_2(\frac{s}{2}, \frac{s}{2})$
- deux élements $(x, s - x)$ et $(s - x, x)$ sinon.

Génération Aléatoire

Solution Méthode en dimension 2: $RandomSourceVectorK2(s,t)$

- Si la fonction T_2 est différentiable
  - Il est possible d'utiliser la méthode de Newton pour résoudre

$$T_2(x, s-x) = t$$

- Sinon on utilise une recherche dichotomique.
- Si $x < \frac{s}{2}$, on renvoie $(x, s-x)$ ou $(s-x, x)$ avec probabilité $\frac{1}{2}$.
- Si $x = \frac{s}{2}$, on renvoie l'unique solution $(x, s-x)$.

Génération Aléatoire: dimension 3

Algorithme 2 : RandomSourceVectorK3

Entrées : Une cible t, un entier k, une valeur $0 < s \leq 1$

Sorties : Un vecteur source modifié π de dim. k tel que $T_k(\pi) \sim t$, ou une erreur

$x \leftarrow$ Tirer aléatoirement une valeur **valide** selon s et t;

$s' = s - x$;

$t' \leftarrow Update_{3,1}(t, x)$;

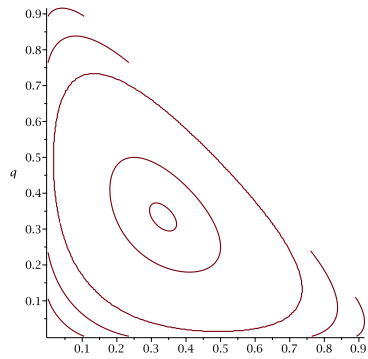
$(y, z) \leftarrow RandomSourceVectorK2(s', t')$;

if $y \neq z$ ou avec probabilité $\frac{1}{2}$ **then**

  return (x, y, z)

end

return Une erreur



En dimension supérieure: Chaîne de Markov

En dimension $k > 3$ on utilise une méthode basée sur les chaînes de Markov.

- On commence par produire un élément de $\mathbb{S}_{k,\varepsilon,t}(s)$
- On effectue ensuite une marche aléatoire sur $\mathbb{S}_{k,\varepsilon,t}(s)$
- On montre qu'il existe que la distribution stationnaire est la distribution uniforme.

En dimension supérieure: Chaîne de Markov

En dimension $k > 3$ on utilise une méthode basée sur les chaînes de Markov.

- **On commence par produire calculer un élément de** $\mathbb{S}_{k,\varepsilon,t}(s)$

$$\exists x \in]0,1[, \exists y, z \in]0,1[, (\underbrace{x, \ldots, x}_{k-2}, y, z) \in \mathbb{S}_{k,\varepsilon,t}(s)$$

En dimension supérieure: Chaîne de Markov

En dimension $k > 3$ on utilise une méthode basée sur les chaînes de Markov.

- On commence par produire calculer un élément de $\mathbb{S}_{k,\varepsilon,t}(s)$
- **On effectue ensuite une marche aléatoire sur** $\mathbb{S}_{k,\varepsilon,t}(s)$
  - On passe d'une solution à une autre en changeant trois variables parmi k
  - On peut utiliser le générateur pour $k = 3$

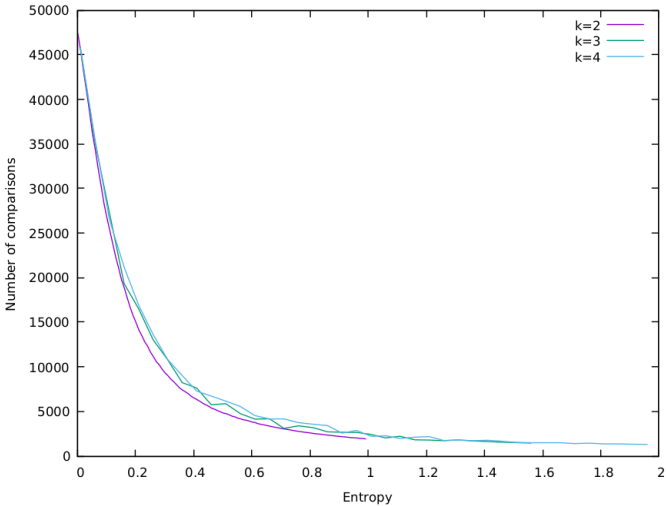
En dimension supérieure: Chaîne de Markov

En dimension $k > 3$ on utilise une méthode basée sur les chaînes de Markov.

- On commence par produire calculer un élément de $\mathbb{S}_{k,\varepsilon,t}(s)$
- On effectue ensuite une marche aléatoire sur $\mathbb{S}_{k,\varepsilon,t}(s)$
- **On montre qu'il existe que la distribution stationnaire est la distribution uniforme. Pour cela on montre que la chaîne est:**
  - irréductible
  - apériodique
  - réversible

Algortihme Naïf: Benchmarks

Algorithme naïf: entropie de Shannon



Ici $n = 1000$ et $m = 50$

- $t = 0$: Pire cas
- $t = \log_2(k)$: cas moyen uniforme
- t fixé: coût moyen augmente avec k

Algorithme naïf: entropie de Shannon

Théorème Complexité moyenne

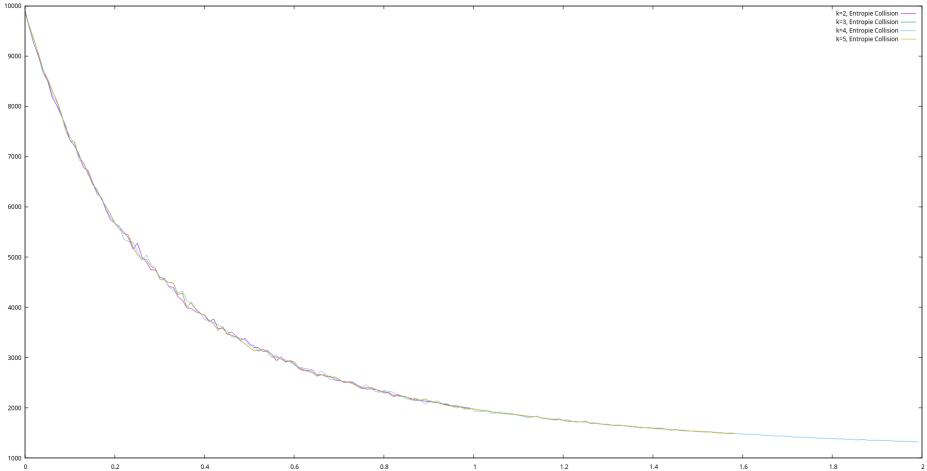
Supposons que $k = 2$ et que le texte de longueur n et le motif de longueur m aient été engendré par une source commune d'entropie t. Alors la complexité moyenne de l'algorthme naïf est

$$(n - m + 1)\left(1 + \frac{y_t - y_t^m}{2(1 - y_t)} + \frac{1 - y_t - (1 - y_t)^m}{2y_t}\right)$$

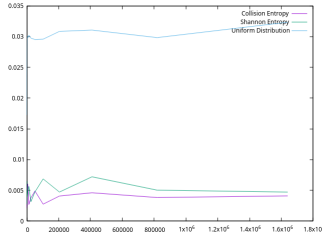
avec $y_t = ie(t)^2 + (1 - ie(t))^2$ et $ie(t)$ est la solution de l'équation

$$T_2(x, 1 - x) - t = 0$$

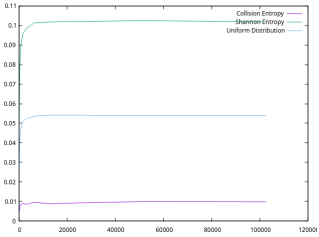
Algorithme naïf: entropie de collision



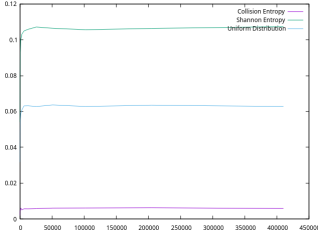
Taux d'erreur de prédiction des modèles



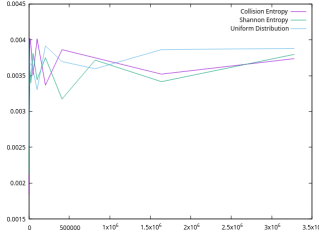
Chromosome 21 ($k = 4$)
Shannon Entropy : 1.9575
Collision Entropy : 1.91818



Hamlet ($k = 88$)
Shannon Entropy : 4.39142
Collision Entropy : 4.23783



Tout du monde en 80 jours ($k = 78$)
Shannon Entropy : 4.34095
Collision Entropy : 3.93015



Ecoli ($k = 4$)
Shannon Entropy : 1.99982
Collision Entropy : 1.99964

Résultat théorique

Théorème Complexité moyenne

Supposons que $k \geq 2$ et que le texte et le motif ont été engendrés par des sources d'entropie de collision commune t, alors la complexité moyenne de l'algorithme naïf appliqué sur un texte de longueur n et un motif de longueur m est

$$(n - m + 1)\frac{1 - 2^{-t \times m}}{1 - 2^{-t}}$$