Vulgariser la méthode de *Cross-Entropy*

De la résolution du Mastermind à l'admission d'étudiants dans des universités

Kim Antunez[*] Alain Quartier-la-Tente[†]

Résumé La *Cross-Entropy* est une méthode adaptée pour résoudre des problèmes d'optimisation combinatoire. Nous avons implémenté l'algorithme en langage R et l'avons ensuite appliqué sur deux cas pratiques : la résolution du Mastermind et la répartition d'étudiants dans des universités.

Mots-clefs : Statistique – Optimisation combinatoire

Cross-Entropy et résolution de problèmes d'optimisation combinatoire

La méthode de *Cross-Entropy* est une méthode utilisée pour estimer des probabilités d'événements rares. Elle peut aussi être adaptée pour résoudre des problèmes d'optimisation combinatoire et c'est dans ce cadre que nous l'exploitons dans cette présentation.

Le problème est le suivant : soit $\mathcal{X}$ un ensemble fini d'états et S une fonction de score. L'objectif est de trouver le maximum de S sur $\mathcal{X}$ (noté γ^*) et les points pour lesquels ce maximum est atteint (x^*). Si on note γ^* ce maximum, on cherche donc :

$$S(x^*) = \gamma^* = \max_{x\in\mathcal{X}} S(x) \tag{1}$$

Pour le résoudre, on lui associe un problème stochastique d'estimation de probabilité d'événements rares. On définit :

- un ensemble d'indicatrices $1_{\{S(x)\geq\gamma\}}$ sur $\mathcal{X}$ pour plusieurs seuils $\gamma \in \mathbb{R}$;
- $\{f(\cdot; v),\, v \in \mathcal{V}\}$ une famille de probabilités sur $\mathcal{X}$, paramétrée par un paramètre vectoriel v.

Pour $u \in \mathcal{V}$, le problème 1 est équivalent au problème d'estimation de la probabilité :

$$\mathbb{P}_u(S(X) \geq \gamma) = \sum_x 1_{\{S(x)\geq\gamma\}} f(x; u) = \mathbb{E}_u[1_{\{S(x)\geq\gamma\}}]$$

L'algorithme utilisé est alors le suivant :

1. Initialisation : on fixe arbitrairement $\hat{v}_0$, deux paramètres $N \in \mathbb{N}$ et $\rho \in]0, 1[$, $t = 1$.

2. On génère un échantillon $X_1, \ldots, X_N$ de loi $f(\cdot, v_{t-1})$, on calcule le quantile $(1-\rho)$ de la fonction score qui donne $\hat{\gamma}_t = S_{\lceil(1-\rho)N\rceil}$. Si $\hat{\gamma}_t \geq \gamma^*$, on prend $\hat{\gamma}_t = \gamma^*$.

3. On utilise le même échantillon $X_1, \ldots, X_N$ pour trouver $\hat{v}_t$:

$$\hat{v}_t = \underset{v}{argmax}\ \hat{D}(v) = \underset{v}{argmax} \frac{1}{N}\sum_{i=1}^{N} 1_{\{S(X_i)\geq\hat{\gamma}_t\}} \ln f(X_i; v) \tag{2}$$

   En d'autres termes, $\hat{v}_t$ correspond à l'estimateur du maximum de vraisemblance de $\hat{v}_{t-1}$ basé sur les ρ meilleures réalisations en termes de score.

4. Arrêt : si pour un certain $t \geq d$, (par exemple $d = 5$), on a : $\hat{\gamma}_t = \hat{\gamma}_{t-1} = \cdots = \hat{\gamma}_{t-d}$

[*] Insee, kim.antunez@ensae.fr
[†] Insee, alain.quartierlatente@ensae.fr
avec la participation de Romain Lesauvage pour les travaux sur le Mastermind.

Application 1 : Mastermind

Le Mastermind est un jeu à deux joueurs. Le premier joueur choisit un *code* : une séquence de n boules de couleur (valeur standard $n = 4$), parmi m couleurs possibles (valeur standard $m = 6$). Puis, le second joueur doit deviner ce code en un minimum de coups. À chaque coup, le second joueur propose un code, et le premier joueur doit lui donner le nombre de boules bien placées (boules noires) et mal placées (boules blanches)

En s'inspirant de la démarche de Rubinstein and Kroese (2004), notre objectif a alors été d'appliquer la méthode de *Cross-Entropy* pour résoudre un Mastermind avec les paramétrisations suivantes :

$$\mathcal{X} = \{1, 2, \ldots, m\}^n$$

$$S(x) = \frac{\omega_{noir} \times N_{\text{boules noires}} + \omega_{blanc} \times N_{\text{boules blanches}}}{\omega_{noir} \times n} \qquad \text{avec} \quad \omega_{noir} > \omega_{blanc}$$

$$\mathcal{V} = \left\{ (p_{i,j})_{i,j} \in \mathcal{M}_{n,m}([0,1]) \ : \ \forall i, \sum_{j=1}^{m} p_{i,j} = 1 \right\} \qquad \text{avec} \qquad \hat{v}_0 = \left(\frac{1}{m}\right)_{i=1..n, j=1..m}$$

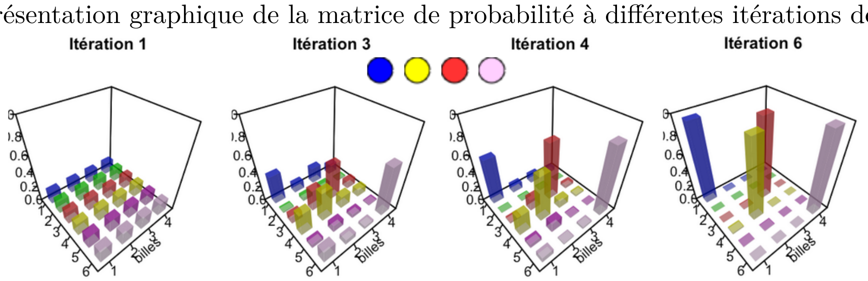
La maximisation de la vraisemblance permet d'obtenir cette formule pour le calcul des paramètres optimaux

$$p_{k,l} = \frac{\sum_{i=1}^{N} 1_{\{S(X_i) \geq \hat{\gamma}_t\}} 1_{\{X_{i,k}=l\}}}{\sum_{i=1}^{N} 1_{\{S(X_i) \geq \hat{\gamma}_t\}}}$$

Simulations et résultats

Une application web permettant d'observer le résultats de différentes simulations est disponible à l'adresse suivante : https://antuki.shinyapps.io/mastermind.

Figure 1: Représentation graphique de la matrice de probabilité à différentes itérations de l'algorithme



Application 2 : Admissions des élèves dans des universités

Nos expérimentations en cours portent sur l'application de la méthode de *Cross-Entropy* à des algorithmes de *matching* qui concernent, en particulier, la répartition d'élèves dans des universités. Quels sont les avantages et inconvénients de la méthode de *Cross-Entropy* comparativement à des algorithmes de matching classiques comme celui de Gale and Shapley (1962) ?

La réponse à cette question vous sera présentée aux rencontres R du mois de juillet !

Références

Gale, David, and Lloyd S Shapley. 1962. "College Admissions and the Stability of Marriage." *The American Mathematical Monthly* 69 (1): 9–15.

Rubinstein, Reuven Y., and Dirk P. Kroese. 2004. *The Cross Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag.