

MDI230 - PROJET VÉLIB

JUIN 2021

1. PRÉSENTATION

On considère un système de vélos partagés, type Vélib, où les vélos sont disponibles dans des stations dédiées et peuvent être empruntés pour faire des trajets d'une station à une autre. Le but de ce projet est de calculer les probabilités stationnaires que chaque station soit vide. Pour cela, il faut :

- Proposer une modélisation du Vélib
- Ecrire un simulateur du processus de Markov correspondant pour 5 stations
- Calculer la probabilité stationnaire théorique dans le cas d'un seul vélo
- Comparer au résultat de votre simulation
- Simuler le même réseau avec 100 vélos.

2. MODALITÉS

Le travail peut se faire en monôme, binôme, ou trinôme. Les groupes ne peuvent pas changer d'un projet à l'autre. Vous pouvez faire la simulation dans le langage de votre choix, nous vous conseillons Python ou Matlab.

- Avant le 1er juin : vous devez lire document sur le modèle des colonies disponible sur le Moodle.
- Le **1er juin à 10h15**, nous vous proposons une séance Zoom pour vous présenter le projet et le travail à faire
- Vous devez déposer sur le Moodle pour le **1er juin à 18h00** un fichier contenant votre modèle du Vélib (correspondant à la section 3 de ce document)
- Le soir du **1er juin à 19h05** sera disponible sur le Moodle la modélisation que nous souhaitons que vous étudiez. Ainsi que les paramètres et formules nécessaires à la calibration du modèle.
- Vous devez déposer sur un fichier pdf contenant les réponses aux questions théoriques (voir ci-dessous) et les résultats obtenus pour la simulation avec 100 vélos pour le **21 juin 2021 à 23h59**. Le code peut-être en annexe. Il vous est recommandé de travailler en Python Notebook puis d'exporter la feuille de calcul en pdf.

3. MODÉLISATION

On suppose que la capacité des stations est illimitée pour l'accueil des vélos, c'est-à-dire que l'on peut toujours rendre un vélo dans une station. On suppose aussi que tous les trajets se font d'une station à une autre exclusivement, il n'est pas possible de revenir à la même station.

1. Lisez le document sur le modèle des colonies disponible sur le site pédagogique (extrait du livre *Lectures on stochastic networks* par F. Kelly et E. Yudovin).
2. Inspirez-vous de ce modèle pour proposer une modélisation du Vélib.
 - Donnez l'espace d'états, et les colonies de votre modèle.
 - Tracez le diagramme des transitions possibles entre les colonies (limiter le dessin à quelques stations)
 - Indiquez les taux de transitions.
 - Donnez les paramètres de votre modèle, précisez les unités.

Date limite : 1er juin 2021 à 18h00

4. SIMULATION

On propose d'étudier le Vélib de Rouen dans sa configuration de 2016 qui possédait 21 stations. Pour la simulation, on ne conserve que 5 stations (stations 3 à 7), les paramètres calculés sont donnés dans le fichier Excel *Donnees_simulations.xlsx* mis à disposition sur le Moodle, ou ci-dessous. Vous pouvez utiliser les conditions initiales que vous le souhaitez, des conditions initiales possibles sont données dans le fichier excel.

Station	3	4	5	6	7
3		3	5	7	7
4	2		2	5	5
5	4	2		3	3
6	8	6	4		2
7	7	7	5	2	
Temps moyen de trajet τ_{ij} en minutes					
Station	3	4	5	6	7
	2,8	3,7	5,5	3,5	4,6
Taux de départ par heure λ_{di}					
Station	3	4	5	6	7
3		0,22	0,32	0,2	0,26
4	0,17		0,34	0,21	0,28
5	0,19	0,26		0,24	0,31
6	0,17	0,22	0,33		0,28
7	0,18	0,24	0,35	0,23	
Matrice de routage p_{ij}					

3. Simulez les trajectoires du processus de Markov.
4. En déduire la probabilité que chaque station soit vide après 150 heures.
5. Calculer l'intervalle de confiance de ce résultat.

5. CALCUL THÉORIQUE

Une fois écrit votre simulateur, qui sortira forcément des résultats, il faut valider votre code par des résultats théoriques.

6. Utilisez les équations de trafic pour obtenir les relations entre les α_i (notations du document sur les colonies).
7. On considère qu'il n'y a qu'un seul vélo, quelle est alors la taille de l'espace d'état ?
8. Dans ces conditions (un seul vélo), calculer la probabilité que chaque station soit vide.
9. Comparez aux résultats obtenus par simulation.

Pour résoudre numériquement le système d'équations donnant les α_i , on suggère de mettre le système sous forme de calcul matriciel $M\alpha = X$ où α est le vecteur colonne des α_i . On a alors $\alpha = M^{-1}X$. Pour éviter $X = 0$, on pourra remplacer une ligne de M par des 1, on a en effet $n + 1$ équations pour n inconnues si on ajoute la condition de normalisation.

6. SIMULATION POUR 100 VÉLOS

10. Simuler le même réseau avec 100 vélos, initialement répartis de façon uniforme entre les stations et les routes.
11. Calculer la probabilité stationnaire que chaque station soit vide par la méthode décrite en section 7.4. On n'oubliera pas de préciser l'intervalle de confiance.

Date limite : 21 juin 2021 à 23h59

7. AIDES POUR LA SIMULATION

7.1. Accélération de votre simulation. Si vous voulez accélérer notablement votre simulation, vous pouvez utiliser la commande `@jit` de la librairie *numba* avant la définition de chaque fonction.

```
1 from numba import jit
2
3 @jit
4 def pickState(p):
5     r=np.cumsum(p)
6     r =r/np.sum(p)
7     u=np.random.rand()
8     w=0
9     while(u>r[w]):
10
11         w+=1
12     return w
```

LISTING 1. Simulation d'une variable aléatoire discrète à valeurs dans $0, 1, \dots, N$ où N est la longueur du vecteur de probabilité passé en paramètre

7.2. Simulation d'une variable aléatoire discrète. Le code ci-dessus met en œuvre l'algorithme de simulation qui suit. Soit X une variable aléatoire discrète et $E = \{x_1, x_2, x_3\}$ son espace d'états. On note $p_i = \mathbf{P}[X = x_i]$ pour $i = 1, 2, 3$ avec $\sum_{i=1}^3 p_i = 1$. On tire une variable aléatoire u uniforme sur $[0, 1]$, alors :

- Si $u \leq p_1$, $\hat{X} = x_1$ et $\mathbf{P}[\hat{X} = x_1] = p_1$,
- Si $p_1 < u \leq p_1 + p_2$, $\hat{X} = x_2$ et $\mathbf{P}[\hat{X} = x_2] = p_1 + p_2 - p_1 = p_2$,
- Si $p_1 + p_2 < u$, $\hat{X} = x_3$ et $\mathbf{P}[\hat{X} = x_3] = 1 - (p_1 + p_2) = p_3$.

Ainsi, \hat{X} a la même loi que X .

7.3. Intervalles de confiance. On rappelle que l'intervalle de confiance d'une quantité $\theta = \mathbf{E}[X]$ estimée sur n tirages X_1, \dots, X_n de moyenne $\hat{\theta}_n = \frac{1}{n} \sum_{i=1}^n X_i$ est l'intervalle de la forme $[\hat{\theta}_n - \varepsilon, \hat{\theta}_n + \varepsilon]$ dans lequel on est sûr à $\alpha\%$ que se trouve θ . On a $\varepsilon = \beta \sigma_n / \sqrt{n}$ avec :

- σ_n l'écart-type empirique sur les tirages :

$$\sigma_n^2 = \frac{1}{n-1} \sum_{j=1}^n (X_j - \hat{\theta}_n)^2 = \frac{1}{n-1} \sum_{j=1}^n (X_j)^2 - \frac{n}{n-1} \hat{\theta}_n^2.$$

Si les X_j valent 0 ou 1, alors $X_j^2 = X_j$ et,

$$\sigma_n^2 = \frac{n}{n-1} (\hat{\theta}_n - \hat{\theta}_n^2).$$

- $\beta = 1,96$ pour $\alpha = 0,95$.

7.4. Simulation régénérative. On sait que la probabilité stationnaire d'une chaîne de Markov irréductible, récurrente peut s'écrire

$$\pi(y) = \frac{1}{\mathbf{E}\tau_x^1} \mathbf{E}_x \left[\sum_{j=0}^{\tau_x^1-1} \mathbf{1}_{\{X_j=y\}} \right]$$

où τ_x^1 est l'instant de premier retour en x partant de x . De manière plus générale

$$\bar{z} = \sum_{y \in E} f(y) \pi(y) = \frac{1}{\mathbf{E}\tau_x^1} \mathbf{E}_x \left[\sum_{j=0}^{\tau_x^1-1} f(X_j) \right].$$

Simulons R cycles de x à lui-même, soit (d_1, \dots, d_R) leur longueur, i.e. $d_j = \tau_x^j - \tau_x^{j-1}$. Soit

$$Y_l = \sum_{j=\tau_x^{l-1}}^{\tau_x^l} f(X_j).$$

Les $((d_k, Y_k), k \in \{1, \dots, R\})$ sont indépendants et identiquement distribués donc

$$\begin{aligned} \hat{\tau}_R &= \frac{1}{R} \sum_{k=1}^R d_k \xrightarrow{\text{p.s.}} \mathbf{E}_x[\tau_x] = \bar{\tau} \\ \hat{Y}_R &= \frac{1}{R} \sum_{k=1}^R Y_k \xrightarrow{\text{p.s.}} \mathbf{E}_x \left[\sum_{j=0}^{\tau_x-1} f(X_j) \right] = \bar{y} \\ \sqrt{R} \left((\hat{\tau}_R, \hat{Y}_R) - (\bar{\tau}, \bar{y}) \right) &\xrightarrow{\text{loi}} \mathcal{N}(0, \Gamma) \end{aligned}$$

avec

$$\Gamma = \begin{pmatrix} \text{var}(\tau_x^1) & \text{cov}(\tau_x^1, Y_1) \\ \text{cov}(\tau_x^1, Y_1) & \text{var}(Y_1^2) \end{pmatrix}.$$

Un estimateur consistant mais biaisé de \bar{z} est donné par

$$\hat{z}_R = \frac{\hat{Y}_R}{\hat{\tau}_R}.$$

La précision de cet estimateur est donné par

$$\sqrt{R}(\hat{z}_R - \bar{z}) \xrightarrow{\text{Loi}} \mathcal{N}(0, \eta^2)$$

où

$$\eta^2 = \gamma_{11} \frac{\bar{y}^2}{\bar{\tau}^4} - 2\gamma_{12} \frac{\bar{y}}{\bar{\tau}^3} + \gamma_{22} \frac{1}{\bar{\tau}^2}.$$

L'intervalle de confiance à 95% est donc donné par

$$\bar{z} \in]\hat{z}_R - \frac{1,95 \eta_R}{\sqrt{R}}, \hat{z}_R + \frac{1,95 \eta_R}{\sqrt{R}}[.$$

Comme on ne connaît pas a priori ni les γ_{ij} , ni \bar{y} , ni $\bar{\tau}$ on les remplace par leur version « empirique » : $\bar{\tau}$ est remplacé par $\hat{\tau}_R$, \bar{y} est remplacé par \hat{Y}_R et γ_{ij} par

$$\hat{\gamma}_{12}^R = \frac{1}{R-1} \sum_{k=1}^R d_k Y_k - \hat{\tau}_R \hat{Y}_R$$

$$\hat{\gamma}_{11} = \frac{1}{R-1} \sum_{k=1}^R d_k^2 - \hat{\tau}_R^2$$

$$\hat{\gamma}_{22} = \frac{1}{R-1} \sum_{k=1}^R Y_k^2 - \hat{Y}_R^2$$