

Modèles et algorithmes des réseaux

DM : date limite 19 janvier 2018

Exercice 1. Apprentissage dynamique des routes

Le but de cet exercice est de simuler un réseau de files d'attente avec apprentissage dynamique des routes.

Le réseau est représenté par un graphe pondéré $G = (V, E, w)$ avec, pour chaque sommet $u \in V$:

- une file d'attente f_u adoptant une politique FIFO ;
- une table de routage $r_u : V \rightarrow V$ telle que $(u, r_u(v)) \in E$ pour tout $v \in V$ (il s'agit du " next hop " pour aller à v) ;
- des arrivées de paquets depuis l'extérieure selon un processus de Poisson de paramètre λ , où chaque paquet p est doté d'une destination $d_p \in V$, choisie uniformément parmi $V \setminus \{u\}$ lorsqu'il entre dans le système ;
- un taux de service $1/w((u, r_u(d_p)))$, où p est le paquet en tête de file (i.e. le temps de service moyen est le poids de l'arête le long de laquelle le prochain paquet est transmis).

Lorsque le paquet p est servi par la file f_u , si $r_u(p_d) = p_d$, i.e. si le paquet arrive à destination, celui-ci quitte le réseau, sinon il est transmis à la file $f_{r_u(p_d)}$.

Question 1 Simuler un tel réseau avec les paramètres suivants :

- G est uniformément distribué sur les 4-réguliers connexes sur 30 sommets ;
- $\lambda = 1$ et les poids des arêtes sont i.i.d., uniformes sur $[0.1, 0.5]$;
- les tables de routages choisissent les plus courts chemins pondérés.

Calculer empiriquement le temps de séjour moyen d'un paquet dans le réseau.

On s'intéresse désormais au cas où le poids des arêtes change au fil du temps, et les tables de routage sont adaptées en conséquence :

- pour chaque arête e , $w(e)$ est modifié selon un processus de Poisson de paramètre μ ;
- les nouveaux poids des arêtes sont tirés selon une même loi W , indépendamment du poids précédent ;
- chaque table de routage r_u est actualisée individuellement selon un processus de Poisson de paramètre ν .

En plus de la valeur $r_u(d)$ pour chaque destination d , chaque table de routage maintient un score $s_u(d)$ correspondant au poids estimé du chemin jusqu'à d (à noter que $s_u(u)$ est toujours égal à 0). Lorsque la table r_u est actualisée, on choisit pour chaque destination d un nouveau $r_u(d)$ correspondant au voisin v_d de u qui minimise $w((u, v)) + r_v(d)$. $s_u(d)$ est alors remplacé par $w((u, v_d)) + s_{v_d}(d)$.

Question 2 En reprenant la simulation précédente, incorporer les modifications présentées ci-dessus, en prenant comme paramètres $\mu = 0.01$, $\nu = 0.1$, et W uniformément distribué sur $[0.1, 0.5]$. Calculer à nouveau le temps de séjour moyen d'un paquet dans le réseau.

Exercice 2. Critère de Foster

On suppose un système de deux files d'attente, q_1 and q_2 , donné dans la figure 1. Le temps est discret

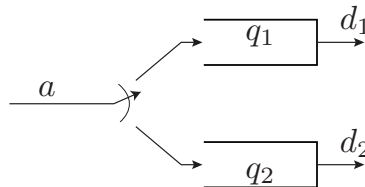


FIGURE 1. Système avec deux files.

et à chaque instant la probabilité qu'un paquet arrive dans le système est égale à a . A son arrivée, le paquet est dirigé vers la file la plus courte. Si le nombre de paquets est égal dans les deux files, le paquet est envoyé dans la file 1. Chaque file a un seul serveur et si la file i n'est pas vide, un paquet est servi et

quitte la file avec la probabilité d_i . Le paquet arrivé dans à l'instant t peut être servi dans le même slot du temps (les arrivées sont considérés avant les services). Soit $X_i(t)$ le nombre de paquets dans la file i au temps t . Alors, l'évolution du système est donné par :

$$X_i(t+1) = X_i(t) + A_i(t) - D_i(t), \quad i \in \{1, 2\},$$

où $A_i(t)$ dénote le nombre de paquets arrivés dans la file i et $D_i(t)$ les départs de la file i à la date t .

Q1. Montrer que $X(t) = (X_1(t), X_2(t))$, $t \in \mathbb{N}$ est une chaîne de Markov. Supposons que $a = 0.4$ et $d_1 = d_2 = 0.25$. Alors la chaîne de Markov est récurrente positive. Cependant, la fonction $V(x) = x_1 + x_2$ ne vérifie pas le critère de Foster. Expliquer pourquoi.

Q2. Est ce que les fonctions suivantes vérifient le critère de Foster ?

(a) $V_a(x) = (x_1 + x_2)^2$,

(b) $V_b(x) = (x_1 - x_2)^2$,

(c) $V_c(x) = x_1^2 + x_2^2$.

Pour chaque fonction, justifier votre réponse.

Q3. Sous quelles conditions sur a , d_1 et d_2 la chaîne est récurrente positive ?

Exercice 3. Une file d'attente

Nous allons considerer une file d'attente M/G/1, avec les arrivées qui suivent un processus de Poisson de paramètre λ (le temps moyen entre les deux arrivées est $1/\lambda$) et un seul serveur. Les temps de service sont indépendants et identiquement distribués selon une distribution générale. Notons par S la variable aléatoire représentant le temps de service (le temps moyen de service est $E[S]$ et le taux moyen de service $1/E[S]$). On suppose que $P(S > 0) = 1$ (i.e. on peut avoir un seul départ à la fois). Les services sont indépendants des arrivées. Notons par $N(t)$ le nombre de paquets dans le système à la date t , par $R(t)$ la variable aléatoire représentant le temps résiduel dans le service (si la file est vide, alors $R = 0$), et $W(t)$ le temps d'attente (le temps entre l'arrivée dans la file et le début de service) pour un paquet qui arrive à la date t . On note par $\rho := \lambda E[S]$ (l'intensité de trafic). On va supposer que $\rho < 1$.

1. Notons par $X_q(t)$ le nombre de paquets en attente de service à la date t . Nous avons donc,

$$E[W(t)] = E[X_q(t)]E[S] + E[R(t)].$$

Montrer que à l'équilibre

$$E[R] = \frac{\lambda E[S^2]}{2}.$$

En déduire que le temps moyen d'attente vérifie :

$$E[W] = \frac{\lambda E[S^2]}{2(1-\rho)},$$

2. Montrer que le nombre moyen de paquets dans le système vérifie :

$$E[X] = \lambda E[S] + \frac{\lambda^2 E[S^2]}{2(1-\rho)}.$$

Exprimer le temps moyen de paquets en fonction du ρ dans le cas ou le service est déterministe (file M/D/1). Comparer par rapport aux cas d'un service exponentiel (file M/M/1).

3. Notons par N_k la taille de la file d'attente juste après le départ du client k , et par V_k le nombre de paquets qui sont arrivées pendant le temps de service de paquet k . Montrer que $\{N_k\}_k$ est une chaîne de Markov en temps discret. Montrer que cette chaîne de Markov est stable si $\rho < 1$.

4. Notons par $\hat{N}_k = (N_k - 1)^+$. Alors $N_{k+1} = \hat{N}_k + V_k$. Notons par $N, \hat{N}, V$ les limites de $N_k, \hat{N}_k, V_k$ et par $G_N, G_{\hat{N}}, G_V$ leur fonction génératrices ($G_N(z) := E[z^N], \dots$). Montrer que

$$G_{\hat{N}}(z) = \frac{G_N(z) - (1-\rho)(1-z)}{z},$$

et

$$G_V(z) = S^*((1-z)\lambda),$$

avec $S^*(s) := E[e^{-sS}]$. Par exemple, si $S \sim \text{Exp}(\mu)$ alors $S^*(s) = \frac{\mu}{s+\mu}$.

5. En déduire que

$$G_N(z) = \frac{(1-\rho)(1-z)}{S^*((1-z)\lambda) - z} S^*((1-z)\lambda).$$

6. En utilisant les résultats précédents, exprimer la transformée de Laplace T^* pour le temps moyen de séjour dans le système en fonction du S^* . Montrer que pour une file M/M/1, on obtient

$$T \sim \text{Exp}(\mu - \lambda).$$

7. Notons par B la durée pendant laquelle le serveur est occupé (le temps entre l'arrivée d'un paquet dans une file vide et le premier instant quand la file devient de nouveau vide). Montrer que

$$E[B] = \frac{E[S]}{1 - \rho}.$$

8. Montrer que

$$B^*(s) = S^*(s + \lambda - \lambda B^*(s)).$$

(Aide : montrer une relation de récurrence pour B en examinant ce qui se passe pendant le service du premier paquet.)

Exercice 4. Réseaux avec pertes

On s'intéresse à un réseau dans lequel le trajet de chaque flux de données est déterminé à l'avance et la capacité nécessaire pour transmettre ces données est réservée elle aussi à l'avance. Ce type de réseaux s'appelle un réseau à commutation de circuit et est approprié pour modéliser un (vieux) réseau téléphonique. On suppose que le réseau a J liens de communication et que la capacité du lien j est C_j . Sur ce réseau, un ensemble de routes R est défini, où chaque $r \in R$ est un sous-ensemble de J . Pour passer un appel en utilisant la route r , il faut réserver une unité de ressource sur chacun des liens empruntés par r . Si la capacité nécessaire n'est pas disponible sur au moins un des liens concernés au moment où un appel arrive, l'appel est rejeté. Des appels arrivent pour la route r suivant un processus de Poisson d'intensité ν_r et chaque appel a une durée $\text{Exp}(1)$; les temps inter-arrivées et les durées des appels sont tous indépendants les uns des autres.

- (1) Décrire la chaîne de Markov représentant le système et calculer sa probabilité stationnaire.
- (2) Quelle est la probabilité de blocage d'un appel sur la route r ?

Exercice 5. Choix de routage par fonction d'utilité

L'approche de partage de ressources en utilisant des fonctions d'utilité peut aussi être appliquée au routage (ou même au problème joint de routage et contrôle de congestion). On considère un réseau constitué de liens L sur lequel des utilisateurs U veulent transmettre des données. Chaque utilisateur $u \in U$ veut transmettre à un taux total y_u et a pour cela à sa disposition un ensemble de routes spécifiques R_u , dont chacune est un sous-ensemble de L . L'utilisateur u répartit donc son trafic en envoyant à un taux x_r sur chaque route r , tel que $\sum_{r \in R_u} x_r = y_u$. On note f_l le flot total sur le lien $l \in L$, i.e. $f_l = \sum_{r \ni l} x_r$.

On va s'intéresser dans cet exercice à minimiser le coût total pour transporter le débit y : on définit pour chaque lien l une fonction $D_l(f_l)$ qui représente le coût au niveau du lien l pour transporter une quantité de trafic f_l . On suppose que D_l est une fonction convexe, croissante, continûment dérivable sur $\mathbb{R}_+$. Le problème que l'on se pose est donc le suivant :

$$\begin{aligned} \min \quad & F(x) = \sum_{l \in L} D_l(f_l) \\ \text{s.t.} \quad & \sum_{r \in R_u} x_r = y_u, \quad \forall u \in U \\ & f \geq 0. \end{aligned}$$

1. Démontrer la propriété suivante :

Propriété 1. Soit F une fonction convexe définie sur un ensemble convexe Ω et de gradient continu ∇F . On a $x^* \in \arg \min_{x \in \Omega} F(x)$ ssi $\nabla F(x^*) \cdot (x - x^*) \geq 0$ pour tout $x \in \Omega$.

2. Démontrer la propriété suivante :

Propriété 2. Un vecteur de trafic admissible x pour la demande y minimise le coût total $F(x)$ ssi il existe $\lambda \in \mathbb{R}_+^U$ tel que $\sum_{l \in r} D'_l(f_l) \geq \lambda_u$, pour tout $u \in U$ et tout $r \in R_u$, avec égalité si $x_r > 0$.

La propriété 2 suggère d'utiliser l'algorithme de déviation de flots suivant : à chaque itération,

- on calcule le trafic f_l sur chaque lien $l \in L$ et la dérivée du coût $D'_l(f_l)$;
- pour chaque route r , on fait remonter à la source les dérivées du coût sur les liens. Cela permet d'obtenir λ_u pour chaque utilisateur $u \in U$;
- pour un certain $\alpha \in [0; 1]$, chaque utilisateur dévie une fraction α du trafic de chaque route non localement optimale vers la meilleure route.

3. Ecrire les équations différentielles décrivant approximativement l'évolution du système pour α petit.
4. En déduire la convergence de cet algorithme vers une solution du problème.

Exercice 6. Calcul distribué de majorité

Soit $G = (V, E)$ un graphe non-orienté. Supposons que la valeur initiale des nœuds soit 0 ou 1. Les nœuds ont une mémoire qui consiste de 2 bits uniquement. On cherche à calculer la valeur majoritaire par un algorithme distribué.

Au temps t , les valeurs de nœuds $x_k(t), k \in V$, appartiennent à l'ensemble $\{0, 0.5-, 0.5+, 1\}$ (ces quatre valeurs différentes peuvent être codées sur 2 bits). Nous allons supposer un ordre total suivant : $0 \leq 0.5- \leq 0.5+ \leq 1$. Un nœud est choisi de manière uniforme parmi tous les nœuds. Ce nœud choisit de manière uniforme un de ces voisins. Notons par i le nœud avec l'état le plus petit parmi ces deux nœuds et l'autre par j . Les nœuds i et j mettent à jour leurs valeurs selon la règle suivante (pour l'addition, on ignore les $-$ et les $+$ pour les valeurs $0.5-$ et $0.5+$) :

$$\begin{aligned} x_i(t+1) &= \left\lfloor \frac{x_i(t) + x_j(t)}{2} \right\rfloor \\ x_j(t+1) &= \left\lceil \frac{x_i(t) + x_j(t)}{2} \right\rceil \end{aligned}$$

Quand $x_i(t+1)$ ou $x_j(t+1)$ est égal à 0.5, alors on doit spécifier s'il s'agit de $0.5-$ ou $0.5+$. Nous allons considérer quatre cas différents :

- Si $x_i(t) = x_j(t)$, alors $x_i(t+1) = x_j(t+1) = x_i(t)$,
- Si $x_i(t) \neq x_j(t)$ et $x_i(t+1) = x_j(t+1) = 0.5$, alors $x_i(t+1) = 0.5+$ et $x_j(t+1) = 0.5-$.
- Si uniquement $x_i(t+1) = 0.5$, alors $x_i(t+1) = 0.5-$.
- Si uniquement $x_j(t+1) = 0.5$, alors $x_j(t+1) = 0.5+$.

Nous allons montrer quelques propriétés de cet algorithme.

1. Montrer les propriétés suivantes :

(a) Préservation de la moyenne :

$$x_i(t+1) + x_j(t+1) = x_i(t) + x_j(t).$$

- (b) Contraction : les valeurs de $x_i(t+1)$ et $x_j(t+1)$ sont soit égales soit consécutives. De plus, si $x_i(t) = x_j(t)$, alors $x_i(t+1) = x_j(t+1) = x_i(t) = x_j(t)$.
- (c) Si $x_i(t) \leq x_j(t)$, alors $x_i(t+1) \geq x_j(t+1)$. En particulier, si les valeurs de $x_i(t)$ et $x_j(t)$ sont consécutives, alors elles sont échangées : $x_i(t+1) = x_j(t)$ et $x_j(t+1) = x_i(t)$.
- (d) $\min(x_i(t), x_j(t)) \leq x_k(t+1) \leq \max(x_i(t), x_j(t)), k \in \{i, j\}$.

2. En utilisant les propriétés précédentes, montrer que cet algorithme converge au sens suivant : après un temps presque sûrement fini, les valeurs des nœuds sont soit toutes égales, soit il y a uniquement deux valeurs différentes et ces deux valeurs sont consécutives (au sens de l'ordre total $0 \leq 0.5- \leq 0.5+ \leq 1$).

De plus, si au temps $t = 0$ la valeur majoritaire est 1, alors ces deux valeurs consécutives sont $0.5+$ et 1. Par symétrie, si la valeur majoritaire est 0, les seules valeurs seront 0 et $0.5-$. Si au temps $t = 0$ il y a autant de 0 que de 1, alors après un temps presque sûrement fini, on aura que des $0.5-$ et des $0.5+$.

3. Proposer un algorithme de type gossip à mémoire finie qui répond au problème suivant. Supposons que la valeur initiale des nœuds soit 0 ou 1. On cherche à vérifier si au moins $2/3$ des valeurs initiales sont égales à 1. Quelle est la taille de mémoire nécessaire pour votre algorithme ?

Exercice 7. Jeu de tennis

Un joueur de tennis a un service rapide (R) et un service lent (L). Les probabilités que le service soit valide sont respectivement p_R et p_L . Si le service est bien valide, les probabilités de gagner le point sont respectivement q_R et q_L . Nous allons supposer que $0 < p_R < p_L < 1$ et $1 > q_R > q_L > 0$. Le problème consiste à trouver la meilleure stratégie en fonction de la situation (score actuel, 1er ou 2ème service) afin de maximiser la probabilité de remporter le jeu.

1. Formuler ce problème comme un problème du plus court chemin stochastique.
2. Montrer que toute politique est propre.
3. Donner les équations de Bellman.

4. Exercice de programmation. On suppose des valeurs suivantes : $q_R = 0.6$, $q_L = 0.4$, $p_S = 0.95$. Donner la courbe de la probabilité que le serveur gagne le jeu en utilisant la stratégie optimale en fonction de la valeur de p_R (avec les increments de 0.05).

5. Supposons maintenant que

$$\frac{p_R q_R}{p_L q_L} > 1.$$

Montrer que la stratégie optimale consiste à toujours utiliser le service rapide.

6. Supposons maintenant que

$$\frac{p_R q_R}{p_L q_L} < 1 + p_R - p_L.$$

Montrer que la stratégie optimale consiste à toujours utiliser le service lent.

7. Décrire le mieux possible la stratégie optimale si on n'est pas dans ces deux cas.