Éléments de processus stochastiques

Projet 2 - Processus continus et files d'attente

Bloc 3 du Bachelier en Sciences informatiques

Profs. Yvik Swan et Pierre Geurts

Année académique 2016-2017

Ce projet vise à mettre en pratique différents concepts relatifs aux processus de Markov en temps continu et à valeurs discrètes. La première partie du projet se focalise sur la simulation de processus de Poisson simple. Dans le seconde partie, on se propose d'étudier et d'optimiser, toujours à l'aide de simulations, un système de file d'attente multi-serveur.

Ce travail est à réaliser par groupe de 2 étudiants. Il devra être rendu au plus tard pour le **vendredi 19/05/2017 à 23h59** par mail à `p.geurts@ulg.ac.be`. Le mail envoyé doit suivre les consignes suivantes :

- Le sujet du message doit être « [MATH1222][projet-2]NOM1-NOM2 » où NOM1 et NOM2 correspondent aux noms des personnes du groupes dans l'ordre alphabétique.
- Dans le corps du mail, les noms, les prénoms et les matricules des personnes constituant votre groupe doivent être clairement indiqués.
- Il doit être joint une archive au format .zip contenant votre rapport et le code permettant de répondre aux questions. L'archive doit être nommée comme le sujet du mail.
- Le rapport final, au format pdf obligatoirement, ne doit pas faire plus de 10 pages avec une typographie et mise en page adéquate. Autrement dit, le pdf ne contient pas plus de 10 pages au total. En outre, le choix, la taille et le design des graphiques doivent être appropriées pour un rapport scientifique.
- Le travail peut être réalisé dans le langage de programmation de votre choix (avec une préférence pour MATLAB, R, Python, Java ou C).

Les projets seront testés via le système anti-plagiat de l'ULg. En cas de plagiat avéré, le groupe se verra attribué une cote nulle à l'ensemble des projets.

1 Simulation d'un processus de Poisson

Questions :

1. Une manière de générer une variable aléatoire X à valeurs continues selon une distribution donnée est de générer une variable aléatoire uniforme U dans $[0, 1]$ et de calculer $X = F^{-1}(U)$ où F^{-1} est l'inverse de la fonction de répartition F de la loi cible ($F(x) = P(X \leq x)$). Utilisez ce principe pour proposer un générateur pour une loi exponentielle de paramètre λ.

2. En utilisant ce générateur, expliquez comment simuler un processus de Poisson $\{X_t, t \geq 0\}$ de paramètre λ sur un intervalle de temps $[0, T_{max}]$.
3. Utilisez cette approche pour générer et représenter graphiquement l'évolution d'un processus de Poisson de paramètre $\lambda = 4$/heures sur une durée de 12 heures.
4. Sur base de 1000 réalisations de ce même processus sur 24 heures, calculez le nombre moyen d'arrivées sur les deux premières heures et sur les deux dernières heures, ainsi que l'écart moyen entre deux arrivées successives. Expliquez les résultats obtenus sur base de la théorie.

2 Étude d'un système multi-serveur

Un système multi-serveur est composé de 3 serveurs répondant à des requêtes de clients. Les requêtes arrivent selon un processus de Poisson avec un taux d'arrivées moyen d'une requête toutes les 40 secondes. Le temps de service pour les quatre serveurs sont tous modélisés par des variables aléatoires exponentielles de paramètres respectifs $\lambda_1 = 0.3$, $\lambda_2 = 0.6$, et $\lambda_3 = 0.9$/min. En plus des requêtes en cours de traitement, jusqu'à 7 requêtes peuvent être mises en attente dans une file. Cette file est unique et partagée par tous les serveurs. Tant que la file est complète, un signal 'occupé' est envoyé en réponse à toute nouvelle requête (qui n'est donc pas traitée). Par défaut, chaque requête est assignée à un serveur au hasard parmi ceux qui sont disponibles à leur arrivée dans le système et les requêtes mises en attente dans la file sont traitées selon une politique FIFO.

On se propose d'étudier les propriétés de ce système et de l'améliorer sur base de simulations numériques. On vous demande pour cela d'écrire un programme permettant de simuler ce système sur une certaine durée T_{max} (et suffisamment souple pour répondre aux questions qui suivent). Votre programme devra pouvoir démarrer la simulation soit avec un système totalement vide, soit avec un système totalement plein. Dans ce dernier cas, chaque serveur sera en train de traiter une requête (dont le traitement pourra être supposé avoir démarré au temps 0) et la file sera remplie. Votre programme devra fournir en sortie une liste des requêtes reçues par le système avec pour chacune d'elle, au minimum, son temps d'arrivée, une valeur booléenne indiquant si oui ou non la requête a pu être traitée et, le cas échéant, le serveur qui s'est occupé d'elle, son temps de début de prise en charge et son temps de sortie du système, et le nombre total de requêtes dans le système à l'arrivée de cette requête (cette requête non comprise).

Questions : Sur base de votre programme, on vous demande de répondre aux questions suivantes :

1. Donnez dans votre rapport une table montrant les informations demandées ci-dessus pour les 10 premières requêtes à partir des deux conditions initiales possibles.
2. Générer une réalisation de ce processus à partir des deux conditions initiales possibles et tracez sur un graphe l'évolution de la variable X_t comptant le nombre de requêtes dans le système (c'est-à-dire en train d'être traitées par un des serveurs ou en attente dans la file) à l'instant t.
3. Générez une réalisation suffisamment longue [1] du processus et tracez en fonction du

1. La durée de cette réalisation doit être suffisante pour que les valeurs mesurées convergent, si elles doivent converger.

temps l'évolution des grandeurs suivantes (en mettant les courbes correspondant aux deux conditions initiales sur le même graphe) :

— le temps moyen [2] qu'une requête passe dans le système (calculé uniquement sur base des requêtes n'ayant pas reçu le signal 'occupé'),
— le nombre moyen de requêtes dans le système lors de l'arrivée d'une nouvelle requête (cette requête n'intervenant pas dans le comptage),
— la proportion de requêtes rejetées,
— le pourcentage de requêtes traitées par chacun des serveurs.

Comparez les courbes obtenues dans les deux conditions et expliquez les résultats observés.

4. Calculez le générateur du processus et déduisez-en sa distribution stationnaire. Calculez le nombre moyen de requêtes ainsi que le temps moyen passé par une requête dans le système, en régime stationnaire. Comparez ces résultats avec les résultats de simulation de la question précédente.

5. On souhaite étudier l'effet sur le système précédent de l'introduction d'un temporisateur sur les requêtes : lorsqu'une requête doit attendre plus de 4 minutes dans la file d'attente, elle est retirée automatiquement du système et ne doit plus être traitée. Si elle est prise en charge avant 4 minutes, elle est traitée complètement même si le temps de traitement total de la requête dépasse 4 minutes.

   (a) Expliquez pourquoi cette modification rend impossible la modélisation du système par un processus de Markov.

   (b) Modifiez votre simulateur pour prendre en compte la temporisation et utilisez le pour calculer *en régime stationnaire* les mêmes grandeurs qu'à la question 2.3 :
   — le temps moyen qu'une requête complètement traitée passe dans le système
   — le nombre moyen de requêtes dans le système à l'arrivée d'une nouvelle requête
   — la proportion de requêtes recevant le signal 'occupé' et la proportion de requêtes dont la temporisation a expiré
   — le pourcentage de requêtes traitées par chacun des serveurs

   Comparez ces valeurs à celles obtenues sans la temporisation à la question 2.3. Justifiez de manière intuitive les différences observées

   (c) On aimerait faire passer la proportion totale des requêtes non traitées en dessous de 3% (approximativement). Est-il possible d'atteindre cet objectif en augmentant
   — la taille de file d'attente ?
   — la puissance de l'un des serveurs ?

   Si c'est possible, dans les deux cas, précisez de combien devrait être au minimum l'augmentation pour atteindre cet objectif.

2. La moyenne doit être calculée sur toutes les requêtes non rejetées entre le temps 0 et le temps courant t, pour des valeurs de t croissantes.