R-project Kansrekening en statistiek:

Deel I: Stochastische processen - Wachtrijtheorie

Academie jaar 2023-2024

Read this first:

- In dit project gaan we verschillende stochastische processen simuleren. We gebruiken hierbij het software pakket R. Andere talen zoals python,... zijn niet toegelaten.
- Dit is een groepsproject dat je maakt met 2 personen. Jullie mogen zelf de samenstelling van de groep kiezen.
- Als eindprodukt van deze opdracht verwachten wij een goed geschreven tekst waarbij je voor elke vraag het probleem verduidelijkt en je uitlegt hoe je tot een oplossing gekomen bent.
- Beantwoord de alle (deel)vragen (met werkelijke getallen), en geef ook een korte (kritische) bespreking over deze antwoorden. Integreer hierbij je R-code in het project.
- Enkel wat R-code met wat getallen en commentaar als eindprodukt is onvoldoende ! Gebruik van AI-tools is ook niet toegestaan en zal bestraft worden !
- Het rapport van dit project moeten jullie electronisch indienen voor vrijdag 17 mei 2024 **voor 23u59** bij zowel mij (roel.braekers@uhasselt.be) als bij de assistent (astrid.sierens@uhasselt.be).
- Je verslag mag maximaal 6 pagina's zijn.
- Vragen kunnen gesteld worden gedurende de hele periode.

Opgaven:

1. In de eerste oefening kijken we naar een Markov keten. Beschouw het volgende spelletje:

   Bas en Harry spelen een spel. Op tafel ligt een rij van zes muntjes, waarvan het meest linkse en het meest rechtse met munt naar boven liggen en alle andere met kop. Een spelbeurt bestaat eruit dat ze kijken of de meest linkse munt met kop of met munt naar boven ligt. Als het kop is, dan mag Bas het muntje oppakken en aan de andere kant van het rijtje met kop of munt naar keuze neerleggen. Als het munt is, dan mag Harry deze keuze maken. Kan Bas er altijd voor zorgen dat alle muntjes met munt naar boven komen te liggen? Het spel stopt op dat moment.

   Dit spel kan je beschrijven als een markov keten wanneer je alle $64 = 2^6$ mogelijkheden voor het rijtje muntjes als verschillende toestanden definieert.

Toestand	Muntjes	Toestand	Muntjes	Toestand	Muntjes	Toestand	Muntjes
1	KKKKKK	17	KMKKKK	33	MKKKKK	49	MMKKKK
2	KKKKKM	18	KMKKKM	34	MKKKKM	50	MMKKKM
3	KKKKMK	19	KMKKMK	35	MKKKMK	51	MMKKMK
4	KKKKMM	20	KMKKMM	36	MKKKMM	52	MMKKMM
5	KKKMKK	21	KMKMKK	37	MKKMKK	53	MMKMKK
6	KKKMKM	22	KMKMKM	38	MKKMKM	54	MMKMKM
7	KKKMMK	23	KMKMMK	39	MKKMMK	55	MMKMMK
8	KKKMMM	24	KMKMMM	40	MKKMMM	56	MMKMMM
9	KKMKKK	25	KMMKKK	41	MKMKKK	57	MMMKKK
10	KKMKKM	26	KMMKKM	42	MKMKKM	58	MMMKKM
11	KKMKMK	27	KMMKMK	43	MKMKMK	59	MMMKMK
12	KKMKMM	28	KMMKMM	44	MKMKMM	60	MMMKMM
13	KKMMKK	29	KMMMKK	45	MKMMKK	61	MMMMKK
14	KKMMKM	30	KMMMKM	46	MKMMKM	62	MMMMKM
15	KKMMMK	31	KMMMMK	47	MKMMMK	63	MMMMMK
16	KKMMMM	32	KMMMMM	48	MKMMMM	64	MMMMMM

   Aangezien er telkens slechts één muntje veranderd wordt, bevat de transitie matrix in elke rij op twee plaatsen een niet-nul kans. Bijvoorbeeld: van KMMKMM kan je enkel naar MMKMMK (je behoudt K met een kans p) of MMKMMM (je verandert K in M met een kans 1-p). Deze kans zullen we laten variëren voor elke toestand en op elke rij krijg je mogelijk een verschillende waarde voor deze kans p.

   (a) Stel de transitie matrix van dit probleem op.

   (b) Simuleer een markov keten voor dit spelletje waarbij zowel Bas als Harry in elke beurt een opgegooid muntstuk laten beslissen of ze een K of een M zullen leggen. Hoeveel spelbeurten zijn er gemiddeld nodig om het spel te stoppen (in toestand MMMMM geraken)? Simuleer hiervoor een voldoende aantal keer dit spel (B= 500 of 1000).

   (c) Herhaal deze simulatie, maar nu gaat Bas proberen het spel meer te controleren. Dus, in toestanden waarbij Bas eerst aan de beurt is en door specifieke keuzes zeer snel naar toestand MMMMMM kan raken, gaan we dit modeleren door de kans p gelijk aan 1 te nemen (voorbeeld: KKKMMM $\rightarrow$ KKMMMM $\rightarrow$ KMMMMM $\rightarrow$ MMMMMM). Hoeveel spelbeurten zijn er nu gemiddeld nodig om het spel te stoppen (in toestand MMMMM geraken)?

2. In de tweede oefeningen gaan we een wachtrij systeem simuleren. We vertrekken hierbij van het computer project 7.26 in het handboek (pagina 208). Opdat niet iedereen hetzelfde systeem zou moeten simuleren, gaan we de parameters van de verdelingen wijzigen, als ook de vragen dit moeten beantwoord worden.

   We verdelen de groep in 4 delen (alfabetisch) en voeren de volgende aanpassingen in:

   (a) jongste en oudste groeplid hebben achternaam tussen Arnauts en Joosten:
      - de tussenaankomsttijd is zoals in de oefening.
      - service times verdeeld als in tabel 1 hieronder.
      - beantwoord vragen (a)-(c)-(e)-(g)-(i)-(k).

   (b) jongste en oudste groeplid hebben achternaam tussen Leendertse en Wertelaers:
      - de gemiddelde tussenaankomsttijd is 2.5 minuut
      - service times verdeeld als in tabel 1 hieronder.
      - beantwoord vragen (a)-(b)-(d)-(f)-(h)-(j).

   (c) jongste groeplid heeft achternaam tussen Arnauts en Joosten, en oudste groeplid heeft achternaam tussen Leendertse en Wertelaers:
      - de gemiddelde tussenaankomsttijd is 2.5 minuut
      - service times verdeeld als in tabel 2 hieronder.
      - beantwoord vragen (a)-(c)-(e)-(g)-(i)-(k).

   (d) oudste groeplid heeft achternaam tussen Arnauts en Joosten, en jongste groeplid heeft achternaam tussen Leendertse en Wertelaers::
      - de tussenaankomsttijd is zoals in de oefening.
      - service times verdeeld als in tabel 2 hieronder.
      - beantwoord vragen (a)-(b)-(d)-(f)-(h)-(j).

Tabel 1.

Server	Distribution	Parameters
I	Exponential	$\lambda = 0.5$ min^{-1}
II	Uniform	$a = 5$ min, $b = 10$ min
III	Uniform	$a = 6$ min, $b = 12$ min
IV	Gamma	$\alpha = 7$, $\lambda = 2$ min^{-1}

Tabel 2.

Server	Distribution	Parameters
I	Uniform	$a = 3$ min, $b = 11$ min
II	Gamma	$\alpha = 5$, $\lambda = 3$ min^{-1}
III	Uniform	$a = 6$ min, $b = 12$ min
IV	Exponential	$\lambda = 0.6$ min^{-1}